Sayawu Yakubu Diaba

On cyber security evaluations in smart grid using machine learning

ACTA WASAENSIA 528

Vaasan yliopisto
UNIVERSITY OF VAASA

ISBN 78-952-395-126-6 (print)
78-952-395-127-3 (online)

ISSN 0355-2667 (Acta Wasaensia 528, print)
2323-9123 (Acta Wasaensia 528, online)

URN http://urn.fi/URN:ISBN:978-952-395-127-3

Hansaprint Oy, Turenki, 2023.

ACADEMIC DISSERTATION

To be presented, with the permission of the Board of the School of Technology and Innovations of the University of Vaasa, for public examination on the 13th of December, 2023, at noon.

Article based dissertation of the School of Technology and Innovations at the University of Vaasa in the field of Telecommunication Engineering.

Author	Sayawu Yakubu Diaba https://orcid.org/0000-0002-7910-4026
Supervisor(s)	Professor Mohammed Elmusrati School of Technology and Innovations University of Vaasa
	Professor Miadreza Shafie-khah School of Technology and Innovations University of Vaasa
Custos	Professor Mohammed Elmusrati School of Technology and Innovations University of Vaasa
Reviewers	Professor Timo Hämäläinen Faculty of Information Technology University of Jyväskyla
	Professor Faisal A. Mohamed Elabdli Libyan Authority for Scientific Research Tripoli Libya
Opponent	Professor Juan Manuel Corchado Department of Computer Science and Automatics University of Salamanca Salamanca Spain

Tiivistelmä

Älyverkko pyrkii parantamaan sähköverkon luotettavuutta, turvallisuutta ja tehokkuutta käyttämällä digitaalista tieto- ja ohjausteknologiaa. Kasvava riippuvuus viestintätekniikasta altistaa kuitenkin nämä järjestelmät kyberhyökkäyksille, mikä aiheuttaa merkittäviä kyberuhkia älyverkon saatavuudelle ja toiminnallisuudelle. Kyetäksemme vähentämään tällaisia uhkia, tehokkaat tunkeutumisen havaitsemisalgoritmit ovat ratkaisevan tärkeitä.

Tässä yhteydessä ehdotamme hybridi syväoppimisalgoritmia, joka keskittyy hajautettuihin palvelunestohyökkäyksiin (DDoS) älyverkon viestintäinfrastruktuurissa. Ehdotettu algoritmi yhdistää konvolutionaalisen neuroverkon (CNN) ja portitetun toistoyksikön (GRU) algoritmit tarjotakseen reaaliaikaista analyysia ja tila-arviopohjaisia tekniikoita tehokkaalle ohjausten toteutukselle. Työssä suoritetaan simulointeja käyttäen Kanadan Kyberturvallisuusinstituutin tunkeutumisen havaitsemisjärjestelmän vertailutietojoukkoa.

Tulokset osoittavat, että hybridi syväoppimisalgoritmimme suoriutuu paremmin kuin olemassa olevat tunkeutumisen havaitsemisalgoritmit, saavuttaen vaikuttavan kokonaistarkkuuden 99,7 prosenttia. Teollisuuden koneita valvovien ja ohjaavien valvonta- ja tiedonkeruujärjestelmien (SCADA) yhteydessä tietoliikenneverkkojen haavoittuvuudet voivat johtaa kyberhyökkäyksiin, joissa väärää tietoa tuodaan operatiiviseen verkkoon. Ehdotamme rajoitettuun Boltzmannin koneeseen perustuvaa ja luonnon inspiroimaa juurten etsinnän optimointialgoritmia kyber-hyökkäysten tunnistamiseen ja luokitteluun. Optimoimme dataominaisuuksia tällä algoritmilla ja arvioimme sen suorituskykyä perinteisiä valvotun koneoppimisen algoritmeja, kuten tekoälyä hyödyntävät neuroverkot, konvolutionaaliset neuroverkot ja tuen vektorikoneet, vastaan. Ehdotettu algoritmi päihittää vertailukohteensa tarkkuudessa, toistettavuudessa ja f1 -pisteissä. Lisäksi työssä käsitellään SCADA-järjestelmien tietoturva-aukkoja esittelemällä geneettisesti alustetun muuntavan neuroverkon (GSFTNN) tunkeutumisen havaitsemisalgoritmin. Toisin kuin allekirjoituksiin perustuvat menetelmät, GS-FTNN havaitsee muutokset toiminnallisten mallien perusteella, jotka viittaavat tunkeutujan osallistumiseen verkkoliikenteessä. Ehdotettua algoritmia arvioidaan käyttäen WUSTL-IIOT-2018 ICS SCADA -kyberturvallisuus tietojoukkoa. Työssä osoitetaan sen ylivoimaisuus perinteisiin algoritmeihin, kuten jäännösneuroverkkoihin, toistaviin neuroverkkoihin ja pitkäkestoisiin lyhytaikamuisteihin, verrattuna tarkkuuden ja tehokkuuden suhteen.

Avainsanat: kyberturvallisuus, syväoppiminen,
tunkeutumisenhavaitsemisjärjestelmät, koneoppiminen, älyverkot, valvova tarkastus ja datan keruu

Abstract

The smart grid aims to enhance the electric grid's dependability, security, and efficiency by deploying digital information and control technology. However, the increasing reliance on communication technology exposes these systems to cyber-attacks, posing significant cyber threats to the availability and functionality of the smart grid. To mitigate such threats, effective intrusion detection algorithms are crucial. In this context, we propose a hybrid deep learning algorithm that focuses on distributed denial of service (DDoS) attacks on the communication infrastructure of the smart grid. The proposed algorithm combines convolutional neural network (CNN) and gated recurrent unit (GRU) algorithms to provide real-time analysis and state estimation-based techniques for efficient control implementation. We conduct simulations using a benchmark cyber-security dataset from the Canadian institute of cybersecurity intrusion detection system. The results demonstrate that our hybrid deep learning algorithm outperforms existing intrusion detection algorithms, achieving an impressive overall accuracy rate of 99.7 %.

In the context of supervisory control and data acquisition (SCADA) systems, which monitor and control industrial machinery, communication network vulnerabilities can lead to cyber-attacks introducing false data into the operational network. We propose a restricted Boltzmann machine-based nature-inspired artificial root foraging optimization algorithm for identifying and classifying cyber-attacks to address this issue. We optimize data features using this algorithm and evaluate its performance against traditional supervised machine learning algorithms such as artificial neural networks, convolutional neural networks, and support vector machines. The proposed algorithm outperforms its counterparts in accuracy, precision, recall, and f1 score. Furthermore, we address the security vulnerabilities in SCADA systems by introducing the genetically seeded flora transformer neural network (GSFTNN) intrusion detection algorithm. Unlike signature-based methods, GSFTNN detects changes in operational patterns indicative of intruder involvement. We evaluate the proposed algorithm using the WUSTL_IIOT_2018 ICS SCADA cyber security dataset and demonstrate its superiority over traditional algorithms like residual neural networks, recurrent neural networks, and long short-term memory (LSTM) in terms of accuracy and efficiency.

Keywords: Cyber-security, deep learning, intrusion detection systems, machine learning, smart grid, supervisory control and data acquisition.

ACKNOWLEDGEMENTS

This study was carried out at the University of Vaasa's School of Technology and Innovations, with essential financial support from the Evald and Hilda Nissi Scholarships Foundation and the University of Vaasa.

I want to express my heartfelt gratitude to my adviser, Prof. Mohammed Elmusrati. Your unwavering support, invaluable guidance, and faith in my capabilities have been pivotal in my academic journey. Your insightful counsel during challenging times in my studies and personal life has been a source of great comfort. Professor Mohammed's dedication to fostering my academic and personal growth has been instrumental in my accomplishments. I genuinely appreciate your ability to inspire and motivate me, pushing me to surpass my boundaries. I am most grateful.

I want to express my sincere gratitude to Prof. Miadreza Shafie-khah, my second supervisor, for the invaluable insights and ideas that you have generously shared. Your contributions have been truly instrumental, and I hold them in high regard. I would also like to sincerely thank Prof. Tommi Lehtonen, Prof. Heidi Kuusniemi, Prof. Emmanuel Nzibah, Prof. Andrew Adewale Alola, and Prof. Marcelo Godey for their guidance and support. Thank you all for your significant roles in my journey.

I wish to thank Dr. Raine Hermans, the School of Technology and Innovations dean, for the invaluable provision of scholarships. I am grateful to Prof. Tomi Pasanen, who leads the School of Technology and Innovations, and the entire faculty and staff. Thanks also go to Prof. Timo Mantere, Prof. Tero Valtainen, Prof. Mike Mekkanen, Prof. Petri Välisuo and Janne Koljonen, whose dedicated efforts have contributed to cultivating an exceptional research environment.

To my dissertation reviewers, Professor Timo Hämäläinen and Professor Faisal A. Mohamed Elabdli, your insights, feedback, warm comments and rigorous scrutiny have challenged me to refine my work and delve deeper into the realm of knowledge. Your collective expertise has shaped my research into something I am immensely proud of. I am especially thankful to Professor Juan M. Corchado for

accepting the role of opponents. Thank you for the insightful discussion.

My time at the University of Vaasa has been enriched by the camaraderie and support of my colleagues and fellow students, Mohammed Saffaf, Arshed Iqbal, Mahmoud Elsanhoury, Abdul Hamid, Tajudeen Ola Hassan, Akpojoto Siemuri, Dalbert Zimuzochukwu, Olaitan Fashanu, Francis Oyeyiola, Ahmed Maruf, Lukumanu Iddrisu ; thank you for your time. To Divine Fomenya, Abdul Rauf Hussein, Zakir Hossain, and Manmeet Singh, the cafes, the late-night discussions, the shared triumphs, and the mutual encouragement have turned classmates into lifelong friends. I am grateful for the community that we have built together. To my brother and motivator, Rabiu Alawo, you are indeed a brother!.

An exceptional thanks go to Dr. Virpi Juppo. You helped me regain my true identity with your kind words when external factors tried to alter it. I say metaphorically; you revived the ”eagle-hood” in me. I re-began my Ph.D. journey from that day when I left *Cafe Oskar* with restored confidence and determination. I am immensely grateful.

I want to acknowledge the broader academic community and the University of Vaasa for providing an environment conducive to growth, learning, and innovation. The opportunities I have had here, the resources at my disposal, and the experiences I have gained have been instrumental in shaping the scholar I have become. As I stand on the cusp of a new chapter, I carry forward the title of ”Doctor” and the lessons learned, the friendships built, and the memories cherished. The emotions swirling within me are a testament to the significance of this achievement. My heart is full of gratitude, humility, and immense accomplishment. Thank you, University of Vaasa, for helping me realize my dreams and for allowing me to contribute to the world of knowledge.

To my cherished friends beyond the academic realm, your companionship and the time we’ve shared are invaluable to me. You’ve brought joy and balance into my life, for which I am deeply grateful. A special shoot-out to my football team, Vaasan Pallo-Veikot; Jeremias Ketonen, Pietari Ketonen, Miro Mikael Roukus, and Joel Eino-Pekka Mäkelä; your camaraderie on and off the field has been a sanctuary for me. Amidst the rigors of research, your support and our moments together have

kept me grounded and mentally resilient. Thank you for being an integral part of my journey. I am also grateful to the Vaasa Islamic Society.
Finally, I am forever indebted to my wife, Amishetu Dicko, and son, Israr Yakubu. Thank you for your understanding, and to my family, Shafie, Laila, Safia, Mohammed, Rubatu, Noora, Sikira, and Zahra, I say thank you for your support. To my mom and best friend, Mariam Hussain, your unwavering love and sacrifices have been the bedrock of my journey. Your support, even from afar, has given me the strength to persevere through challenges and celebrate triumphs. This accomplishment is as much yours as it is mine. All praise is due to Allah (AWJ).

Contents

List of Figures

List of Tables

ABBREVIATIONS

4G	Fourth Generation
5G	Fifth Generation
AMI	Advanced Meter Infrastructure
AI	Artificial Intelligence
AUC	Area Under Curve
ANN	Artificial Neural Network
BPTT	Backpropagation Through Time
BRS	Bias Risk Score
CIC-IDS	Canadian Institute of Cybersecurity Intrusion Detection Systems
CIS	Center for Internet Security
CIR	Class Imbalance Ratio
CART	Classification and Regression Trees
CNN	Convolutional Neural Networks
CRNN	Convolutional Recurrent Neural Networks
CTI	Cyber Threat Intelligence
DLP	Data Loss Prevention
DIA	Data Integrity Attacks
DNN	Deep Neural Networks
DoS	Denial of Service
DER	Distributed Energy Resources
DDoS	Distributed Denial of Service
EMS	Energy Management System
FP	False Positive
FN	False Negative
GRU	Gated Recurrent Unit
GMM	Gaussian Mixture Models
GEE	Generalized Estimating Equations
GPRS	General Packet Radio Services
GSF	Genetically Seeded Flora
GSFTNN	Genetically Seeded Flora Transformer Neural Network
GPS	Global Positioning System
GSM	Global System for Mobile
IAM	Identity and Access Management

ICT	Information and Communication Technologies
IDS	Intrusion Detection Systems
IPS	Intrusion Prevention System
IT	Information Technology
IoT	Internet of Things
ISO	International Organization for Standardization
IaaS	Infrastructure-as-a-Service
KNN	K-Nearest Neighbors
LAN	Local Area Networks
LSTM	Long Short-Term Memory
LoRaWAN	Long Range Wide Area Network
NIST	National Institute of Standards and Technology
NSL-KDD	Network Security Laboratory Knowledge Discovery in Databases
OaT	One-at-a-Time
OvO	One-vs-One
OvR	One-vs-Rest
OTA	Over-the-air Attacks
PCC	Pearson Correlation Coefficient
PLC	Power Line Carrier
PCA	Principal Component Analysis
PCS	Process Control Security
PSSS	Power System State Security
RNN	Recurrent Neural Networks
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
RTU	Remote Terminal Units
RES	Renewable Energy Sources
RBM	Restricted Boltzmann Machines
SSL	Secure Socket Layer
SIEM	Security Incident and Event Management
SMS	Smart Meter Security
SE	State Estimator
SARSA	State-Action-Reward-State-Action
SGD	Stochastic Gradient Descent
SVM	Support Vector Machines
SCADA	Supervisory Control and Data Acquisition

TLS	Transport Layer Security
TN	True Negative
TP	True Positive
UEBA	User and Entity Behavior Analytics
SGCPS	Smart Grid Communication Protocol Security
WUSTL-IIoT	Washington University in St. Louis- Industrial IoT
WAN	Wide Area Networks

LIST OF PUBLICATIONS

This thesis represents a comprehensive compilation of meticulously organized peer-reviewed papers that significantly contribute to the advancement of knowledge in the field of cyber security within smart grids. It encapsulates the outcomes of extensive and rigorous research on the subject matter, effectively summarizing the findings. Roman numerals (I-V) citations are utilized throughout the text to reference the corresponding publications.

(I) On the performance metrics for cyber-physical attack detection in smart grid. **Diaba, S.Y.**, Shafie-khah, M. and Elmusrati, M. Published in *Soft Computing*, 26(23), pp.13109-13118., 2022. **doi.org/10.1007/s00500-022-06761-1**

(II) Proposed algorithm for smart grid DDoS detection based on deep learning. **Diaba, S.Y.** and Elmusrati, M. Published in *Neural Networks*, 159, pp.175-184. 2023. **doi.org/10.1016/j.neunet.2022.12.011**

(III) Cyber Security in Power Systems Using Meta-Heuristic and Deep Learning Algorithms. **Diaba, S.Y.,** Shafie-Khah, M. and Elmusrati, M. Published in *IEEE Access*, 11, pp.18660-18672. 2023. **doi:10.1109/ACCESS.2023.3247193**

(IV) SCADA securing system using deep learning to prevent cyber infiltration. **Diaba, S.Y.**, Anafo, T., Tetteh, L.A., Oyibo, M.A., Alola, A.A., Shafie-Khah, M. and Elmusrati, M. Published in *Neural Networks*. 2023. **doi.org/10.1016/j.neunet.2023.05.047**

(V) Risk Assessment of Machine Learning Algorithms on Manipulated Dataset in Power Systems. **Diaba, S.Y.**, Shafie-Khah, M., Mekkanen, M., Vartiainen, T. and Elmusrati, M. Published in *International Conference on Future Energy Solutions (FES 2023)* (pp. 1-5). *IEEE*. **doi: 10.1109/FES57669.2023.10182751**

AUTHOR'S CONTRIBUTION

Publication I: "On the performance metrics for cyber-physical attack detection in smart grid"

The author diligently executed the experimental procedures for all algorithms considered in the study. This phase involved designing the experiments, setting up the necessary environments, collecting and processing data, and conducting rigorous analyses. As the primary investigator and researcher, the author took the lead in writing the primary manuscript. The immediate and second supervisors played integral roles in providing invaluable guidance and support. Their expertise and experience were instrumental in shaping the manuscript and elevating its academic rigor.

Publication II: "Proposed algorithm for smart grid DDoS detection based on deep learning "

The author took the lead in all aspects of the manuscript, from conceptualization and investigation to validation and writing of the initial manuscript. The primary and secondary supervisors played crucial roles, offering invaluable direction and support, significantly contributing to the manuscript's overall quality and academic rigor.

Publication III: "Cyber Security in Power Systems Using Meta-Heuristic and Deep Learning Algorithms"

The author conceptualized the idea and conducted the experimental analysis with respect to all the algorithms considered. The author wrote the primary manuscript. The immediate and second supervisors help in shaping the manuscript.

Publication IV: “SCADA securing system using deep learning to prevent cyber infiltration”

The author played a pivotal role by conceiving the original idea and meticulously conducting the experimental analysis for all the algorithms under consideration. He authored the primary manuscript. Throughout the process, the immediate and second supervisors provided invaluable support and guidance, playing essential roles in shaping the manuscript and elevating its academic merit.

Publication V: “Risk Assessment of Machine Learning Algorithms on Manipulated Dataset in Power Systems ”

The author had the responsibility of conducting the experimental analysis. The author wrote the primary draft of the manuscript. The other authors reviewed and contributed at different stages of the manuscript preparation.

1 INTRODUCTION

The traditional energy system refers to how energy has been generated and distributed for many decades, relying primarily on fossil fuels such as coal, oil, and natural gas (Belkebir et al., 2018). The generated power is transmitted through an intricate grid system comprising electricity substations, transformers, and power lines connecting electricity producers and consumers. The grids are interconnected for reliability and efficiency, generating more prominent and reliable networks that improve energy supply coordination and planning. The electrical grid is an extensive network of high-voltage power lines distributing for thousands of kilometers and low-voltage power lines with distribution transformers connecting electric power to millions of customers (Fang et al., 2012; Merabet et al., 2017). The electric power generation system produces energy used to power transportation, heat buildings, and supply electricity to homes, companies, and industries. ie., small and large utilities. This system has typical characteristics of massive infrastructure, centralized power generation, and unidirectional energy transfer from producers to consumers driving economic development and technological advancements worldwide.

Although it has been a reliable and effective source of energy for humanity, it has also caused significant environmental damage, including air and water pollution. One of the main challenges is its environmental effect, as it contributes to a significant portion of global greenhouse gas emissions, causing climate change. For example, fossil fuels are finite resources, and their extraction and production can negatively affect local communities and the environment. Specifically, they are facing several challenges, including; the increasing demand for electricity from a growing population. The need to integrate more renewable energy sources (RES) into the grid. The grid's aging infrastructure is becoming more vulnerable to outages. The traditional electric network is vulnerable to disruption (Blumsack and Fernandez, 2012) from natural disasters, vandalism, and other threats, leading to power outages and other disruptions to daily life.

As a result of these challenges, many communities and countries are transitioning to more sustainable RES (V. Kumar et al., 2016), such as solar, and wind power (Ahmed et al., 2020). These energy sources are often distributed and can be generated near the point of use (Huang et al., 2012), reducing the need for large-scale infrastructure and enabling greater energy independence and resilience.

The smart grid is an emerging technology that has the potential to revolutionize the way electric power is produced and consumed. Smart grids for energy management are becoming increasingly popular in addressing climate change and energy security issues. They are intelligent systems that use information and communication technologies (ICTs) to improve energy delivery efficiency, reliability, stability, and sustainability (Yigit et al., 2014). It has become the core of the power net-

work due to its vital benefits, such as self-healing, good power quality, a broader electrical market, the involvement of customers, the ability to accommodate diverse generation sources, storage options, and robustness (Espe et al., 2018). The new generation standard of the power grid permits communication between the smart grid domains. i.e. (generation, transmission, distribution, system operators, service providers, customers, and the power market) (Fang et al., 2012; Metke and Ekl, 2010). This is possible with the introduction of information technology (IT) and communication infrastructure in the power network (Dehalwar et al., 2014; Haluk Gözde; M. Cengiz Taplamacioglo; Murat Ari; Hamza Shalaf, 2015).

Networking these domains makes the smart grid network very complex, leading to severe possible security holes in the grid. These risks are linked to its automation, communication, protection and data collection systems (Tuballa & Abundo, 2016). Using computer software and data transmission in the electrical power grid allows for improved management and optimization of the power system, with real-time monitoring, control, and integration of RESs.

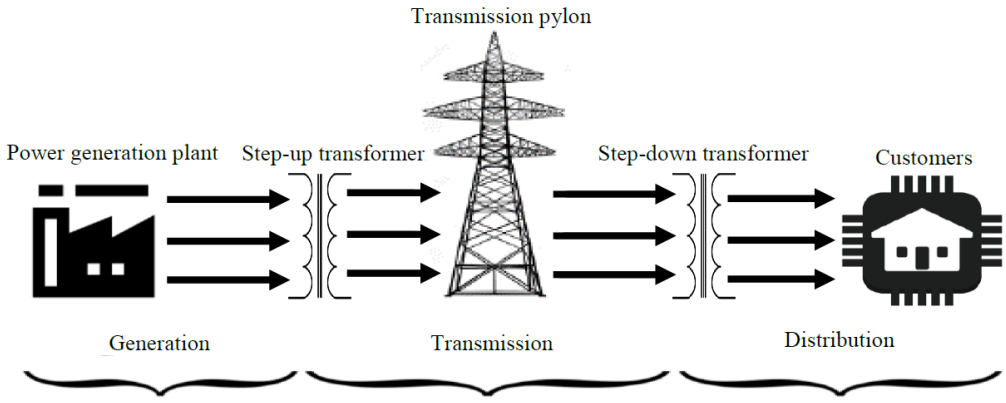


Figure 1: Illustration of traditional power system

Information exchange is vital in ICT systems that are used in the smart grids. This is possible with the aid of the communication network (wireless and wireline) integrated into the power systems. An essential part of smart grids is based on distributed generation (DG) through RES. Hence, wireless communication will offer effective, flexible, and inexpensive solutions for networking. There are several issues associated with wireless networking applications in the protection of power systems, such as latency and reliability. However, with new communication standards such as the 5G networks and beyond, these problems might be handled. The issues of latency and reliability are not discussed in this thesis.

Nevertheless, the vital issue of cyber security will be the main research topic. Since the wireless communication network has its cyber security vulnerabilities, introducing it into the power network would mean tackling the vulnerabilities from

the power network perspective. The smart grid cyber continuation security, in its entirety, covers the privacy and security of the communication and automation of the grid (Diptiben Ghelani, 2022). That facilitates the communication between the smart grid domain and the control of circuit breakers and switches as well as data transmission throughout the power grid.

Vulnerabilities in the power system sense mean the security holes in the network that could be exploited for illegal or unauthorized tasks. These vulnerabilities are mainly; network vulnerability, platform vulnerability, and management vulnerability. Network vulnerability is linked to wireless connections and the communication aspect of the power network. Platform vulnerabilities encapsulate the hardware, the software, the system configuration, and the data-keeping system. Procedure and security standards are considered under management vulnerabilities.

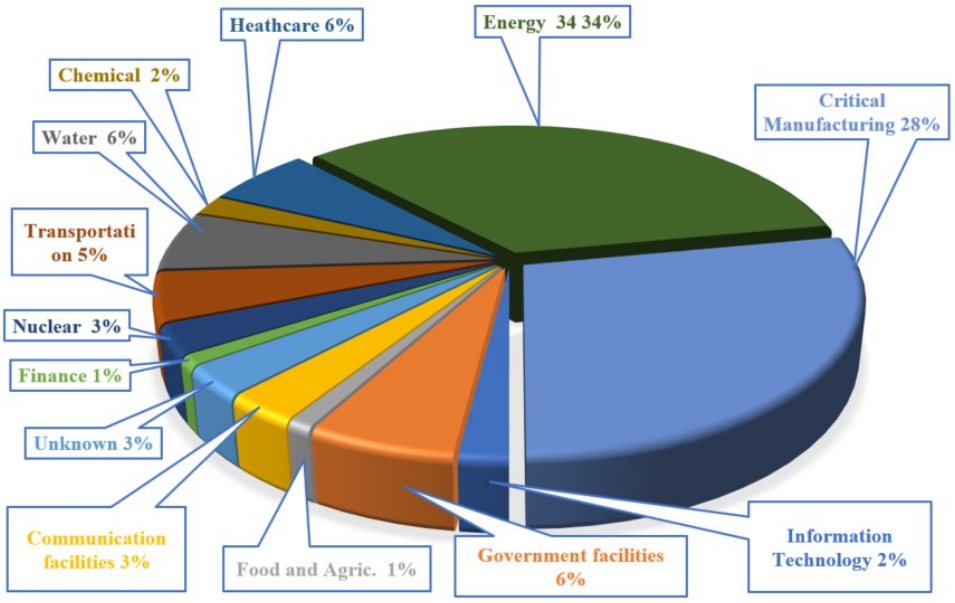


Figure 2: 2014 Fiscal year incidents reported by different sectors (ICS-CERT, 2015).

The energy sector could be the newest target area for cyber-attacks (ICS-CERT, 2015). Therefore, research on cyber attack detection and prevention schemes is necessary. The prevention schemes must be capable of defending the smart grid from cyber-attacks launched by terrorists, dissatisfied or dismissed employees, politicians, industrial spymasters, market competitors, and natural disasters (Yang et al., 2011). Based on the ramification of the smart grid, the security aspect is divided into five categories. Namely: Process control security (PCS), smart meter security (SMS), power system state security (PSSS), smart grid communication protocol

security (SGCPS), and smart grid simulation for security analysis (Diptiben Ghelani, 2022). Looking at different industry sectors, PCS has been used by automated systems to monitor and control processes using computer networks. Yet, it was in isolation and without outside network links. Currently, PCSs are run in connection with other networks from within and outside. This undoubtedly creates holes for cyber attackers to penetrate.

Smart meters are digitalized type of traditional kilowatt-hour (kWh) meter, mainly measuring power, current, and voltage (Avancini et al., 2018). They usually have a programmable microcontroller, memory, and a communication board (e.g., a wireless modem). It might have an output port for remote control as well. This forms the advanced meter infrastructure (AMI). The AMI is arguably the most implemented element (Novosel, 2012) of the smart grid worldwide, as it is seen as the first step for power providers to locomote toward the smart grid. A metering device is classified as smart when it can frequently keep records of the power and other vital parameters and can interconnect this information to the central system for monitoring and analytical purposes. These data can be used to monitor network conditions such as distribution state estimation, advanced distribution operations, transmission operations, and asset management (Hussain et al., 2020).

The AMI communicates to the central system through communication means such as local area networks (LAN), wide area networks (WAN), like LoRaWAN, Bluetooth, Zigbee, and mobile networks (eg., GSM, 4G, 5G, etc.). It can also be communicated through a wireline network such as an Ethernet or power line carrier (PLC). Hence, the AMI implementation establishes the common telecommunications and IT infrastructure (Q. Zhang et al., 2010). These communication requirements include bandwidth, latency, reliability, and security. This means that there will need to be many different protocols used in the smart grid to enable communication between components. The state estimator (SE) has been used in the power sector to monitor, estimate, and control the power network since the evolution of the smart grid. The SE uses the transferred data from all the remote terminal units (RTU) on the network to estimate the true state of the network (Mingkui Wei, 2016). Thus, estimating based on false data will result in bad results by default. Data integrity attacks (DIA) must be defended in all senses.

In order to control and supervise the generation, transmission, and distribution of electricity, an energy management system (EMS) is used. It is a sophisticated software-based system that integrates real-time data and provides monitoring, control, and optimization capabilities to ensure efficient and reliable operation of power systems (Su et al., 2017). EMS typically incorporates supervisory control and data acquisition (SCADA) functionality as a part of its overall architecture (He and Yan, 2016). SCADA systems collect real-time data from various field devices and sensors, such as RTUs, and send that data to a central control center where operators can monitor and control the power system. RTUs are devices deployed at various

points in the electrical grid, such as substations or distribution centers, to measure electrical observations throughout the grid (Oyewole and Jayaweera, 2020). They are equipped with sensors and meters to collect data on various electrical parameters, including voltage levels, current flows, power factors, frequency, and other relevant data points. The RTUs gather real-time data from the field and transmit it to a central control center where operators can monitor and analyze the performance of the electrical grid. They act as intermediaries between the field devices and the central SCADA system.

The measurements telemetered to the data acquisition module of the EMS are used by the SE to provide an estimate of the grid's state and to, monitor and optimize the electric power grid's condition and provide real-time troubleshooting when a critical system component fails. This is illustrated in Figure 3.

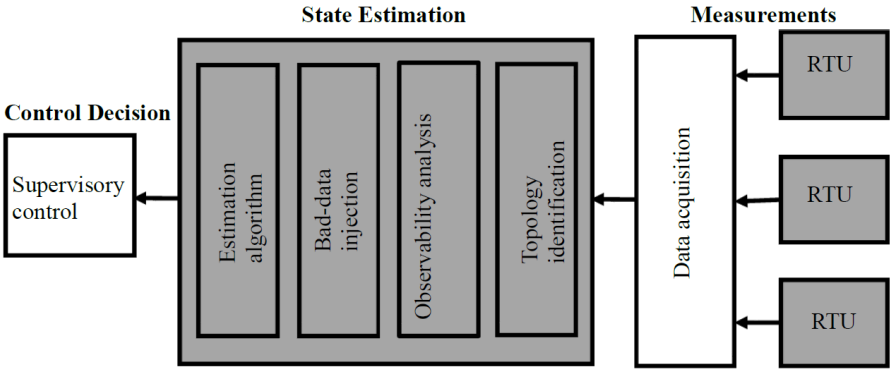


Figure 3: The state estimation within an energy management system

Keeping the smart grid secure is a vital task. However, using very complex protocols and adding complicated layers on the communication part to enhance security will, on the other hand, increase the communication delays between sensors (e.g., AMI) and controllers as well as between the controllers and actuators (e.g., circuit breakers). Such delay might cause colossal damage or operation instability. In some faults, the protection system must react within 5 milliseconds or less to prevent severe damage to the grid. Pre-learned machine learning is one potential technology that can be useful to preserve security and privacy in smart grids.

The application of machine learning in cyber-security has shown promising results in other domains, such as finance, healthcare, and transportation. Therefore, a strong motivation exists to investigate the potential of machine learning-based cyber-security techniques in smart grids. Machine learning can help to detect and predict cyber-attacks in real-time, which can significantly enhance the security of the smart grid. Moreover, machine learning models can learn from historical data

and adapt to new and emerging threats, making them more effective than traditional cyber-security mechanisms. This dissertation aims to evaluate the effectiveness of machine learning-based cyber-security techniques in enhancing the security of the smart grid against cyber-attacks. The research findings provide valuable insights into the potential benefits of using machine learning for cyber-security in the smart grid and contribute to developing more effective and reliable cyber-security mechanisms.

1.1 Problem statement

The smart grid is an evolving power grid infrastructure that integrates advanced communication and information technologies to improve the electricity supply chain efficiency, reliability, and security. However, incorporating these technologies introduces new cyber-security challenges that can compromise the grid's availability, integrity, and confidentiality, potentially leading to catastrophic consequences (Tufail et al., 2021). Traditional cyber-security approaches in the smart grid are insufficient to address emerging threats' complexity and diversity. Therefore, there is a need for advanced cyber-security mechanisms that can leverage the power of machine learning to provide real-time attack detection, prediction, and mitigation.

1.2 Objective

This dissertation aims to evaluate the effectiveness of machine learning-based cyber-security algorithms in enhancing the security of the smart grid against cyber-attacks. It also assesses the risks of trusting machine learning on the network operation and compromising between network efficiency and availability. The specific objectives are:

1. To review the state-of-the-art cyber-security threats and attacks in the smart grid and the traditional cyber-security techniques used to mitigate them.

2. To identify the limitations of traditional cyber-security techniques in addressing the emerging threats and the potential benefits of machine learning-based cyber-security techniques

3. To explore the relevant machine learning algorithms that can be used for real-time threat detection, prediction, and mitigation in the smart grid.

4. To design and implement a machine learning-based cyber-security framework for the smart grid and evaluate its effectiveness in detecting, predicting, and mitigating cyber-attacks.

5. To compare and analyze the performance of the proposed machine learning-based cyber-security algorithms with traditional cyber-security techniques and identify its strengths and weaknesses.

6. To address some future work paths for research continuation in this area, such as risk management issues in AI-based decision-making.

1.3 Outline of the thesis

The remainder of the dissertation is organized as follows: The literature on smart grids and cyber-security is reviewed in detail in Chapter 2. It includes a summary of the smart grid, definition of words related to cyber security, the value of cyber security in the smart grid, and the consequences of cyber security failures in the smart grid. It provides information on the traditional cyber-security measures used by the smart grid as well as the most recent cyber-security threats and assaults. A brief introduction to machine learning is provided in the third chapter, which discusses cyber security in smart grid. The smart grid's cyber-security using machine learning algorithms is given. Decision trees, classification and regression trees, entropy, information gain, Gini index, support vector machines, random forests, deep learning, long short-term memory, convolutional neural networks, recurrent neural network, linear regression, and logistic regression are captured. The chapter highlights machine learning applications in cybersecurity. Risk analysis of machine learning applications, algorithmic risks in machine learning, and algorithmic risk management.
The fourth chapter discusses machine learning methodology for smart grid cyber security. It includes feature extraction, feature selection, data collecting, data cleaning, and preprocessing procedures. It also discusses experimental setup, model selection, model training, hyper-parameter adjustment, and performance evaluation of machine learning algorithms. A model evaluation, sensitivity analysis, risk analysis, and simulation results and discussion are all presented in Chapter 5. In Chapter 6, we present the dissertation conclusions and outline some future research directions in this area.

2 REVIEW OF LITERATURE

This chapter thoroughly assesses existing studies on cyber intrusions in smart grids, with a particular emphasis on using machine learning algorithms in smart grids. The review aims to identify research gaps and propose innovative ways to build on existing knowledge to fill them. The chapter will, thus, contribute to a more thorough understanding of the possible benefits and limitations of utilizing machine learning algorithms for cybersecurity in smart grids by assessing and synthesizing the present state of knowledge. The proposed methodologies will try to fill identified research gaps and provide fresh insights to inform the development of effective smart grid cybersecurity tactics.

2.1 Overview of smart grid

The smart grid is a cutting-edge electrical distribution system that aims to transform how we distribute and use power. It is a modern improvement to the existing power grid that uses advanced technologies and communication networks to improve efficiency, reliability, and sustainability (Lamba et al., 2019; Y. Zhang et al., 2018). This smart grid overview looks into its definition and concept and the advantages and benefits it provides. It also investigates numerous problems and future developments in installing smart grid technologies.

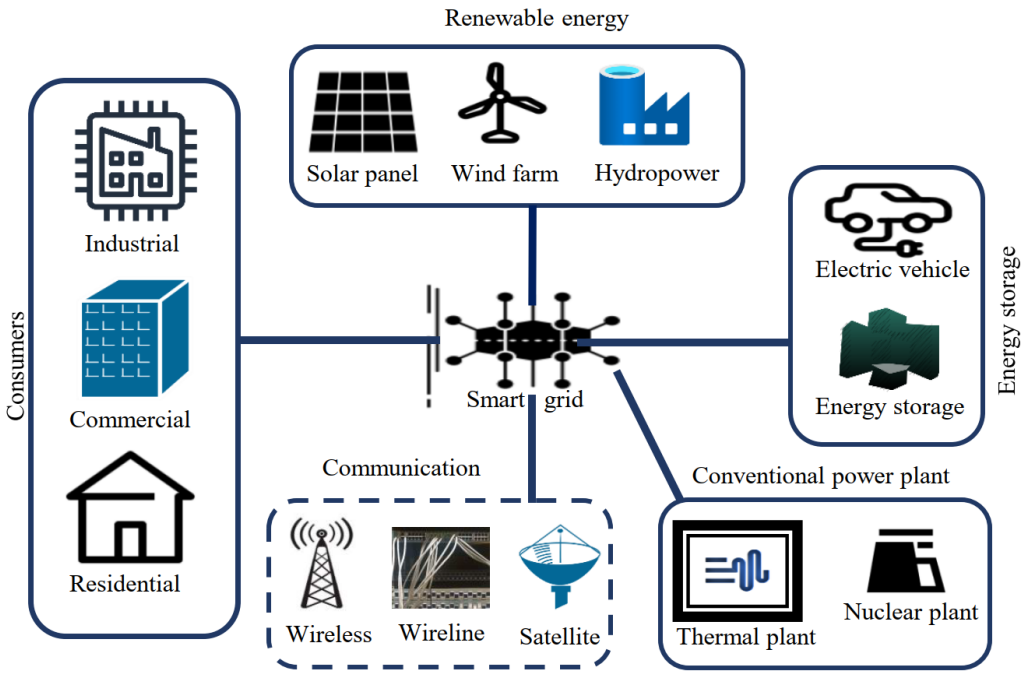


Figure 4: The architecture of smart grids.

A modernized electricity network designed to deliver energy more efficiently and flexibly is shown in Figure 4. According to (Espe et al., 2018), the smart grid is a self-healing, self-optimizing, and self-protecting network that leverages advanced sensors, communication technologies, and control systems to enhance power delivery's reliability, security, and sustainability. One of the critical features of the smart grid is its ability to integrate renewable energy resources (RES) and distributed energy resources (DERs), such as solar panels, wind turbines, and energy storage systems, into the grid. This integration enables more efficient and cost-effective (Sadiq et al., 2021) utilization of renewable energy (Steimer, 2009), reduces greenhouse gas emissions (Mutani et al., 2019), and enhances energy independence. Another important aspect is its capability to enable demand response and dynamic pricing, which allow customers to manage their energy consumption and costs based on real-time information and incentives. It supports advanced meter infrastructure (AMI), which provides utilities and customers with granular information on energy usage, peak demand, and outage management (Youssef et al., 2018).

Smart grid technology can potentially revolutionize the power sector by providing numerous advantages and benefits. In (Hledik, 2009), the paper outlined some key advantages of smart grid technology, such as improved power quality, reduced power outages, and increased energy efficiency. With advanced sensors and communication technologies, it can detect and respond to power outages in real-time, minimizing the impact of outages on customers (Blumsack & Fernandez, 2012; Fang et al., 2012). It is, therefore, a sophisticated and innovative energy system that offers numerous benefits for customers, utilities, and society.

The implementation of smart grid technology has brought about numerous challenges that require attention for the system to function effectively. The challenges include interoperability, cybersecurity, and data privacy (Zerbst et al., 2010). Interoperability challenges arise due to the integration of various communication technologies (Ma et al., 2013) that are not necessarily compatible, leading to data loss and system failures. Cybersecurity threats are also a significant challenge, given that smart grid systems are susceptible to attacks from hackers seeking to exploit vulnerabilities in the system (Kappagantu and Daniel, 2018). Data privacy alarms grow due to smart grid systems collecting vast amounts of data, creating questions about how it is utilized and who has access to it.

Despite all the difficulties and challenges, adopting smart electrical networks is a necessity imposed by reality and necessity. There are several considerable technological breakthroughs addressing these issues. For example, advances in creating secure communication protocols and data encryption technologies have aided in addressing cybersecurity problems. Blockchain technology is being investigated as a tool for improving data privacy and security in smart grid systems (Falahi et al., 2022; Gao et al., 2022; Kuzlu et al., 2020). While smart grid deployment confronts various obstacles, the technology's future appears bright, with advances that solve

these challenges and make the system more efficient and safer.

2.1.1 Cyber-security in smart grid

The smart grid represents a modernized version of the traditional generation, transmission, distribution, and metering infrastructures. This is achieved by upgrading existing systems with digital technologies such as microprocessors, software, and network communication channels. The new technologies can supplement existing components that perform the same function as before, but now provide and communicate information to a centralized system, or offer entirely new functionalities that enable human operators and the grid to respond intelligently to changing conditions (Smith and Pate-Cornell, 2018).

Technology integration in modern power grids has brought about a new era of efficiency and convenience in energy distribution. However, this advancement has also brought about new risks (D. Kumar and S., 2020), particularly in cyber-attacks. As the world increasingly depends on technology, cyber-security detection and prevention have become crucial to ensuring smart grid systems' safety and reliability (Sahu and Davis, 2023). Cyber-security is a critical concern in smart grid technology due to the increased use of interconnected communication systems and devices. Integrating various communication networks and technologies such as sensors, monitoring systems, and control systems creates a complex web of communication susceptible to cyber-attacks (More et al., 2022).

Cyber-attacks on a smart grid can cause catastrophic power outages, resulting in financial losses, environmental harm, and public safety issues. Various procedures must be adopted to ensure the smart grid's cyber-security, including the use of secure communication protocols, firewalls, and intrusion detection systems (IDS) (Baig and Amoudi, 2013). Communication connections and data storage systems must be encrypted to avoid unauthorized access and data breaches. Another critical part of smart grid cyber-security is establishing access controls and authentication procedures to prevent access to vital infrastructure and data networks. Strong passwords, multi-factor authentication, and role-based access restrictions are used to restrict access to authorized individuals exclusively (Diamantoulakis et al., 2015).

The paper (Anwar et al.; 2014) proposed several cybersecurity measures to protect smart grid systems. One of the measures is the implementation of access controls, which restrict access to critical components of the smart grid system. This measure ensures that only authorized personnel can access the system, reducing cyber-attack risk. Another measure is the implementation of IDSs, which monitor and detects any unauthorized access or malicious activity within the smart grid system (Eken, 2013). This measure allows for quick identification and response to potential cyber-attacks. The authors suggested the implementation of encryption technology, which protects smart grid system data from unauthorized access by encrypting it. The

measure ensures that data transmitted through the system is secure and cannot be accessed by unauthorized parties. Implementing these vital cybersecurity measures can protect smart grid systems from cyber-attacks.

In light of the increasing adoption of smart grid technology, cyber-security is a critical consideration in the design and implementation of smart grid technology. Effective cyber-security measures must be put in place to protect against cyber-attacks and ensure the reliable and secure operation of the smart grid. Implementing these measures requires collaboration between stakeholders, including utilities, government agencies, and cybersecurity experts, to develop comprehensive and effective cybersecurity strategies.

2.1.2 Terms of cybersecurity

In the digital era, where technology is deeply integrated into our lives, protecting sensitive information and digital assets has become paramount. Cybersecurity safeguards computer systems, networks, and data from unauthorized access (Ye Yan, Yi Qian, Hamid Sharif, 2013), which has emerged as a crucial discipline in our increasingly interconnected world. Understanding the terminologies associated with cybersecurity is essential for comprehending the concepts, tools, and measures employed to mitigate cyber threats. Cybersecurity involves several crucial elements, including:
Preventive measures: These measures are aimed at preventing cyber-attacks from occurring in the first place. It involves implementing security controls such as firewalls, antivirus software, strong passwords, and encryption to protect against potential threats. **Detective measures**: These measures are focused on identifying and detecting potential security breaches or unauthorized activities. They involve monitoring systems and networks for suspicious activities, analyzing logs, and using IDSs and intrusion prevention systems (IPSs).

Responsive measures: If a cyber-attack occurs or a security breach is detected, responsive measures are taken to mitigate the impact and prevent further damage. Incident response plans, backup and recovery systems, and disaster recovery plans play a crucial role in minimizing the consequences of an attack. **Security awareness and training**: Educating users and employees about potential threats and best practices is essential for maintaining a secure environment. Training programs, workshops, and ongoing awareness campaigns help raise awareness about cybersecurity risks and promote responsible online behavior. **Security policies and procedures**: Establishing robust security policies and procedures is vital for organizations to ensure consistent and effective cybersecurity practices. These policies outline guidelines for the secure use of technology, handling of data, and response to security incidents.

Some key terminologies associated with cybersecurity are:

Malware: Short for malicious software, malware refers to any software designed to cause harm, damage, or gain unauthorized access to computer systems (Gunduz and Das, 2020). It includes viruses, worms, Trojans, ransomware, and spyware. **Phishing**: Phishing is a technique cybercriminals use to trick individuals into providing sensitive information, such as passwords, credit card details, or personal data, by posing as a trustworthy entity. This is typically done through deceptive emails, messages, or websites. **Firewall**: A firewall is a network security device that acts as a barrier between a trusted internal network and an untrusted external network (usually the Internet). It monitors and controls incoming and outgoing network traffic based on predefined security rules to prevent unauthorized access and protect against malicious activities.

Encryption: Encryption is the process of converting information or data into a coded form to deter unauthorized access or interception. It ensures that even if the data is intercepted, it remains unreadable unless decrypted with the appropriate key. **Vulnerability**: A vulnerability refers to a weakness or flaw in a system or software that attackers can exploit to compromise the system's security (Yeboah-Ofori and Islam, 2019). Vulnerabilities can exist in operating systems, applications, networks, or configurations. **Penetration testing**: Also known as ethical hacking or white-hat hacking, penetration testing involves simulating real-world attacks on a system or network to identify vulnerabilities and weaknesses. It helps organizations assess their security posture and address any vulnerabilities before malicious hackers can exploit them.

Two-factor authentication (2FA): Two-factor authentication adds an extra layer of security by requiring users to provide two forms of identification to access a system or account. It typically involves combining a password or PIN with a second factor, such as a unique code generated by a mobile app, a fingerprint scan, or a physical token. **Intrusion Detection System (IDS)** and **Intrusion Prevention System (IPS)**: IDS and IPSs are security mechanisms that detect and respond to potential security breaches. An IDS monitors network traffic and systems for suspicious activities or signs of an attack (Sun et al., 2018). At the same time, an IDS detects and takes active measures to prevent or block such activities. **Data breach**: A data breach occurs when unauthorized individuals access sensitive or confidential data. It can result in the exposure or theft of personal information, financial data, trade secrets, or other valuable information, leading to potential misuse or harm (Y. Li and Liu, 2021).

Social engineering: Social engineering is a tactic used by attackers to manipulate individuals into revealing sensitive information or performing certain actions. It often involves psychological manipulation and deception, exploiting human trust and vulnerabilities to gain unauthorized access to systems or data. These are some of the many terminologies used in the field of cybersecurity. The dynamic nature of cyber threats means that new terms and concepts emerge as the field evolves. Some

more are: **Patch management**: Patch management refers to keeping software and systems up to date with the latest security patches and updates released by vendors. Regularly applying patches helps address known vulnerabilities and strengthens the overall security posture. **Zero-day vulnerability**: A zero-day vulnerability is a security flaw or weakness in a software or system that is unknown to the vendor or developers. Attackers exploit these vulnerabilities before a patch or fix is available, making it challenging for organizations to defend against such attacks. **Data loss prevention** (DLP): Data loss prevention involves implementing measures and technologies to prevent the unauthorized disclosure, leakage, or loss of sensitive or confidential data. DLP solutions monitor and control data flow within the organization and when data is shared with external entities.

Identity and access management (IAM): IAM refers to the policies, processes, and technologies used to manage and control user identities and access to systems, applications, and data. It ensures that only authorized individuals have appropriate access privileges and reduces the risk of unauthorized access (Suicimezov and Georgescu, 2014; Wulf et al., 2019). **Cyber threat intelligence** (CTI): CTI involves gathering and analyzing information about potential and existing cyber threats to identify patterns, tactics, techniques, and indicators of compromise. This information helps organizations proactively detect and respond to cyber threats effectively. **Security incident and event management** (SIEM): SIEM is a centralized security solution that combines security event management and log management to provide real-time monitoring, correlation, and analysis of security events across an organization's network. It helps identify and respond to security incidents and provides valuable insights for security operations.

Some more are, **Bring your own device**: Bring your own device refers to the policy allowing employees to use their personal devices (such as smartphones, laptops, or tablets) for work purposes. While it offers flexibility, it introduces security risks as organizations must manage and secure corporate and personal data on these devices. **Cybersecurity frameworks**: Cybersecurity frameworks provide guidelines and best practices for organizations to establish and improve their cybersecurity programs. Examples include the NIST cybersecurity framework, ISO 27001, and center for internet security (CIS) controls, which provide a structured approach to managing cybersecurity risks. **Secure socket layer/transport layer security** (SSL/TLS): SSL/TLS protocols provide encryption and secure communication over networks, typically used for securing web traffic. They establish a secure connection between a client and a server, ensuring the confidentiality and integrity of the data transmitted (Wulf et al., 2019). Cyber insurance: **Cyber insurance** is insurance coverage that helps organizations mitigate financial losses resulting from cyber-attacks or data breaches. It may cover costs related to incident response, legal liabilities, data recovery, and business interruption.

2.1.3 Importance of cyber security in smart grid

Cybersecurity is an increasingly important topic for any networked informatics system, for example, connected to the internet, and smart grids are no exception. The importance of cybersecurity in smart grids stems from the fact that they are complex systems with many interconnected devices and systems that control the bidirectional flow of electricity. This makes them vulnerable to cyber-attacks, which can have severe consequences in terms of the stability and reliability of the grid (Hahn et al., 2013). Smart grids are designed to be more efficient and reliable than traditional power grids, but they also present new challenges in terms of cybersecurity. With so many devices connected to the grid, there are more potential entry points for attackers to exploit (Mylrea and Gourisetti, 2017). This is particularly true for the software and communication systems that control the grid, which is often the weakest link in the system (Tufail et al., 2021).

To address these vulnerabilities, cybersecurity measures must be integrated into every aspect of the smart grid, from the hardware and software to the communication protocols and data management systems. This requires a multi-layered approach that includes strong encryption, firewalls, IDSs, and regular software updates to patch vulnerabilities (Gunduz and Das, 2020). It also requires training and awareness programs for those who operate and maintain the grid and protocols for responding to cybersecurity incidents. The importance of cybersecurity in smart grids cannot be overstated. A successful cyber-attack on a smart grid could lead to power outages, financial losses, and even physical harm to people and property. It could also impact critical infrastructure and national security (Kimani et al., 2019).

Therefore, it is essential that the cybersecurity of smart grids is taken seriously and that all stakeholders work together to ensure that these systems are as secure as possible. This includes policymakers, regulators, utilities, manufacturers, and cybersecurity experts. By working together, they can protect the smart grid and ensure that it continues to provide reliable and sustainable energy for years to come.

2.1.4 Implications for cyber-security in smart grid

As digitization escalates in the electrical system, so does the number of access points and potential attack paths multiply. Exploitation can be directed at various smart grid components such as software, hardware, and communication networks. A successful cyber-attack will disrupt critical infrastructure (Kimani et al., 2019) such as hospitals, transportation networks, and financial institutions (Diptiben Ghelani, 2022). Cyber-security ramifications in smart grid technologies include the danger of illegal system access, cyber espionage, data breaches, and other types of cyber-attacks. Unauthorized system access is one of the most dangerous consequences of cyber-security in smart grid technology. Smart grid technology mainly relies on communication networks to exchange data between system components.

These communication networks are frequently vulnerable to cyber-attacks, and unauthorized system access can cause severe interruptions in energy distribution. Cybercriminals can exploit system vulnerabilities to access sensitive data, such as consumer information, and manipulate the system to create outages or other damages (Sun et al., 2018).

The use of sensors and monitoring systems to collect data on energy distribution and consumption is tangled in smart grid technology. The data's sensitivity and valuableness make it a target for cybercriminals who seek to use this data to their advantage. Cyber espionage can lead to intellectual property theft, financial losses, and other damage to the energy sector. Data cracks are also a substantial insinuation of cyber-security in smart grid technology. Informed decision-making in the smart grid energy distribution heavily relies upon the data. Any breach of this data can lead to substantial interruptions in the energy supply chain (Liu et al., 2012). Cybercriminals can use data cracks to snip sensitive information, manipulate data, or cause other types of damage to the smart grid system.

Though, integrating smart grid technology has brought momentous benefits to the energy sector. The use of technology in the energy sector also boons several security challenges that must be tackled to ensure the safe and secure operation of the power grid. Cybersecurity is critical for the smart grid system; thus, any compromise of the system can have severe implications for energy consumers and the energy sector (Yan et al., 2012). To mitigate these implications, stakeholders in the energy sector must work together to develop and implement robust cyber-security procedures that can shield the smart grid system from cyber-attacks.

2.2 Traditional cyber-security mechanisms in the smart grid

Traditional cyber-security procedures have been utilized to combat cyber-attacks, but the smart grid provides a distinct set of issues that necessitate specific attention. This section overviews traditional cyber-security methods, emphasizes special smart grid difficulties, and addresses possible remedies and future perspectives for smart grid security. Traditional cyber-security measures include the use of technologies, policies, and processes to prevent unauthorized access, use, disclosure, disruption, modification, or destruction of computer systems.

According to (Cone et al., 2007), these methods include firewalls, IDSs, antivirus software, access control mechanisms, and encryption technologies. Firewalls are crucial in safeguarding private networks against unauthorized access by selectively filtering incoming and outgoing traffic according to predetermined criteria. On the contrary, IDSs are designed to identify and address abnormal network activities, such as exploiting vulnerabilities, spreading malware, or pilfering sensitive data (Workman et al., 2008). Antivirus software detects and eliminates malware from compromised systems, including viruses, worms, and Trojans. Access control mea-

sures like passwords, biometrics, and smart cards are implemented to restrict the access of authorized users solely to sensitive data and applications. Encryption technologies, such as SSL and TLS, ensure data security during transmission and while at rest. These technologies convert data into an unintelligible format that can only be decoded with a confidential key. Although conventional cyber-security measures have successfully mitigated numerous cyber threats, they are not infallible and can be susceptible to sophisticated and persistent attacks that exploit design and implementation vulnerabilities. Consequently, organizations must embrace a comprehensive and forward-thinking approach to cyber-security that integrates traditional measures with emerging technologies and industry best practices.

Smart grid technology presents inherent challenges requiring comprehensive solutions for efficient and effective system operation. As highlighted by the paper (Moslehi and Kumar, 2010), integrating RES within the grid is a prominent concern. The variable and unpredictable nature of renewable sources like wind and solar power can introduce grid instability (Svendsen et al., 2017). Overcoming this hurdle requires implementing sophisticated control systems capable of real-time power supply-demand balancing. Another significant issue revolves around the realm of cyber-security. Given the heavy reliance of the smart grid on communication and information technologies, any compromise in security could lead to severe disruptions in the system. As stated (Moslehi and Kumar, 2010), safeguarding the smart grid necessitates the deployment of critical security mechanisms, including encryption, authentication, and IDSs.

Moreover, handling voluminous data poses a formidable obstacle for the smart grid. Effectively processing, storing, and analyzing the vast amount of generated data is vital in enabling informed decision-making. The paper (Moslehi and Kumar, 2010) propose leveraging modern data analytics techniques such as machine learning and artificial intelligence to tackle this challenge. In conclusion, traditional cyber-security techniques used in smart grids have proven insufficient in protecting against new cyber threats. Cyber-security measures must grow in tandem as the smart grid evolves and becomes more sophisticated. This necessitates transitioning to more advanced and adaptive security technology capable of detecting and responding to new cyber threats in real-time. Furthermore, all players in the smart grid ecosystem must collaborate to create a culture of cyber-security awareness and best practices. Only by taking a thorough and coordinated approach can ensure the safety and security of the smart grid infrastructure.

2.3 Emerging cyber-security threats and attacks in the smart grid

With the advantages that technologies bring to the smart grid, there are also emerging cyber-security threats and attacks that can jeopardize the integrity and security of the smart grid (Procopiou and Komninos, 2015). One of the most concerning

threats is the potential for hackers to gain unauthorized access to the smart grid's control systems. This could lead to power outages, equipment damage, or even physical harm to individuals (Radoglou-Grammatikis and Sarigiannidis, 2019). Another emerging threat is the rise of ransomware attacks which can compromise the data and systems of the smart grid. Ransomware attacks can restrict access to critical data and systems, and attackers can demand payment to restore access (Basnet et al., 2021). Attacks on smart grid infrastructure, such as power substations, could cause widespread power outages and disrupt essential services, such as hospitals, emergency response units, and transportation systems. Phishing attacks are also a significant concern for smart grids. Cybercriminals can use phishing attacks to trick employees or customers accessing the smart grid's systems into providing login credentials or other sensitive information. Once attackers have access to these systems, they can cause significant damage and may even disrupt power distribution. Integrating RES into the smart grid presents unique security challenges (Ayar et al., 2017). Smart grids are vulnerable to new types of cyber-attacks that exploit the vulnerabilities of renewable energy systems (Ding et al., 2022). For example, hackers could target the system's inverters or microcontrollers and cause them to malfunction, leading to power outages.

The paper (Arabo, 2015) highlights some emerging cyber-security threats and attacks in the smart grid, including advanced persistent threats (APTs), insider threats, and supply chain attacks. APTs are sophisticated attacks that use multiple infection vectors and evasion techniques to gain persistent access to the system and steal sensitive information or cause damage. Insider threats are attacks from within the organization, such as disgruntled employees, contractors, or partners with access to critical systems and data (Yang et al., 2011). Supply chain attacks are attacks that target the vulnerabilities of the third-party components and software used in the smart grid, such as communication protocols or sensors, to gain unauthorized access or control over the system. These emerging threats and attacks require new and innovative cyber-security solutions that can detect, prevent, and respond to cyber threats in real-time and improve the resilience and robustness of the smart grid against cyber-attacks. One of the potential threats to the smart grid is the APT attacks (Leszczyna, 2018b). These attacks are difficult to detect and can remain undetected for an extended period, allowing the attacker to access sensitive information and control the system. The insider threat is another significant cyber-security threat to the smart grid. Insiders may abuse their access privileges to compromise the system's confidentiality, integrity, and availability. These threats pose a significant challenge to the smart grid's cyber-security, necessitating the implementation of robust security measures to protect against them (Workman et al., 2008).

According to (Leszczyna, 2018), cyber-security threats and smart grid attacks can come from external and internal sources. External threats may include criminal hackers, nation-states, and terrorist organizations, while internal threats may include disgruntled employees and contractors. The attacks can result in various

consequences, including loss of data, disruption of service, and physical damage to the infrastructure. In order to handle such threats, the energy sector must adopt a comprehensive approach that includes risk assessment (Rohmeyer and Ben-zvi, 2015), cyber-security awareness training, and robust security measures such as firewalls, intrusion detection, and prevention systems. The smart grid's widespread adoption presents significant cyber-security challenges, and the energy sector must take proactive measures to address them to ensure the reliable and secure delivery of energy services (Leszczyna, 2018).

With the benefits of smart grids come new cyber-security threats and attacks. These threats range from simple phishing attempts to more sophisticated attacks, such as malware and denial of service (DOS) attacks. These threats can result in a wide range of negative consequences, including the theft of sensitive data, disruption to the electricity supply, and even physical damage to equipment. As smart grids become more complex, it is becoming increasingly challenging to protect against these threats. One way to address this issue is to implement a multi-layered approach to cyber-security, including technical and non-technical measures. Technical measures include firewalls, IDSs, and encryption, while non-technical measures include training and awareness programs for employees and customers. By taking a comprehensive approach to cyber-security, smart grid providers can help to mitigate the risks associated with emerging threats and attacks (Eder-Neuhauser et al., 2017). The paper (Kimani et al., 2019) highlight some emerging cyber-security threats and attacks in the smart grid, including phishing attacks, ransomware attacks, and DOS attacks. Phishing attacks involve cybercriminals sending fraudulent emails to utility companies or smart grid customers to steal sensitive information. On the other hand, ransomware attacks involve hackers encrypting utility company data, making it inaccessible until a ransom is paid. DOS attacks involve hackers flooding the smart grid system with traffic, causing it to crash (Ashok et al., 2017). These attacks can have severe consequences, including power outages, financial losses, and even loss of life.

Consequently, utility companies must implement robust cyber-security measures to protect the smart grid from cyber-attacks. This can include regular cyber-security training for employees, implementing firewalls and IDSs, and regularly testing the system for vulnerabilities. With the continued evolution of technology, utility companies need to remain vigilant and proactive in mitigating cyber-security threats in the smart grid (Kimani et al., 2019). As technology advances and the grid becomes increasingly interconnected, it is crucial to stay vigilant, continually update defenses, and adopt proactive measures to mitigate these risks.

With a more interconnected and digitally reliant future, these advancements come a new frontier of cyber-security threats and attacks that loom ominously over the stability and reliability of the smart grid. We delve into this landscape of potential chaos and visualize the vivid threats that cast a shadow over the smart grid. These

are examples of emerging cyber-security threats and attacks in the smart grid.

APTs: APTs are sophisticated and long-lasting cyberattacks designed to infiltrate and compromise the smart grid infrastructure. Attackers obtain unauthorized access, remain undetected for an extended period, and target specific assets or systems to disrupt operations or steal sensitive data (Gunduz and Das, 2020). **Ransomware Attacks**: Ransomware has become a prevalent menace across many industries, including the smart grid. Attackers can use ransomware to encrypt vital files and systems, effectively holding the grid infrastructure hostage until the ransom is paid. These attacks can disrupt power generation, distribution, and management systems, resulting in extensive blackouts. Attacks on the supply chain smart grid systems rely on a complex supply chain involving multiple vendors and suppliers. By injecting malicious code or compromising hardware components, attackers can exploit the vulnerabilities in this chain (Yeboah-Ofori and Islam, 2019). This enables unauthorized access and manipulation of the smart grid infrastructure. **Threats from insiders:** Insiders with privileged access to the smart grid infrastructure can pose serious risks. Malicious employees or contractors may abuse their positions to intentionally compromise vital systems, pilfer sensitive data, or disrupt operations. It emphasizes the need for effective access controls and monitoring mechanisms.

Zero-day exploits: A zero-day exploit refers to a cyberattack that takes advantage of a software vulnerability that is unknown to the software vendor or developer. Attackers exploit these vulnerabilities before the software creator becomes aware of them, which means there is ”zero days” of advanced notice to patch or remedy the issue. **Phishing and social engineering**: Cybercriminals frequently use phishing to deceive employees or users into divulging sensitive information or granting unauthorized access. They may pose as vendors or colleagues to acquire trust and manipulate individuals into compromising the security of the smart grid (Tony Flick, 2010). **Malware and botnets**: Malware, including viruses and botnets, can infect smart grid infrastructure devices. These infected devices join a larger network, enabling remote control and coordinated attacks. Malware can disrupt operations, compromise data, and facilitate further intrusion. **DoS and distributed denial of service (DDoS) attacks**: DoS and DDoS attacks overwhelm targeted systems or networks with a deluge of traffic, rendering them inaccessible or degrading performance. These attacks can disrupt communication channels, compromise monitoring systems, and cause the failure of vital smart grid infrastructure components (Yang et al., 2011).

Alongside insider threats, there is also concern regarding insider subversion. Individuals within the organization who cause damage or disruption to the smart grid infrastructure on purpose. Whether motivated by personal benefit, ideology, or vengeance, insider sabotage can have devastating effects, resulting in prolonged power outages or system failures. **AMI attacks**: Smart grids frequently rely on AMI, which consists of smart meters installed in residences and businesses to mon-

itor and control energy consumption. These meters communicate with the grid and provide optimization-relevant data. In addition, they introduce new attack vectors. Attackers may attempt to compromise or manipulate smart meters, resulting in erroneous readings, fraudulent billing, or targeted assaults against specific customers or regions (Tony Flick, 2010). **Global positioning system (GPS) spoofing and time synchronization attacks**: The exact timing and synchronization of devices within the smart grid are essential to its correct operation. Using spoofing techniques, attackers can target and manipulate GPS signals used for time synchronization. This can result in inaccurate time references, disrupting the coordination and synchronization of vital grid operations (Gunduz and Das, 2020). Such disruption in time synchronization can lead to serious protection problems and large damage to the power grid.

Vulnerabilities of cloud-based infrastructure: Smart grids increasingly utilize cloud-based platforms and services for data storage, analytics, and remote management. These cloud environments, however, introduce new vulnerabilities. Attackers may target cloud infrastructure, leveraging misconfigurations or insufficient access controls to obtain unauthorized access to vital grid data or control systems. **Attacks using artificial intelligence and machine learning**: As artificial intelligence and machine learning technologies find applications in the smart grid for optimization and automation; they also become potential attack targets. The integrity and efficacy of artificial intelligence models may be jeopardized if adversaries manipulate or contaminate training data sets. This can result in erroneous decisions, cascading failures, and even targeted attacks against grid systems propelled by artificial intelligence. The capabilities of artificial intelligence can be leveraged to orchestrate severe security breaches against smart grids. In these instances, the conventional methods employed for preventing and detecting cyber-security attacks may fall short of providing adequate protection.

Firmware and hardware tampering: The firmware and hardware components within the smart grid infrastructure are potential targets for tampering. Attackers may modify or replace the firmware in devices or embedded systems, introducing malicious code or backdoors that can compromise the entire system's security. Similarly, compromised or counterfeit hardware components can pose significant risks to the grid's integrity and functionality (Voas, 2016). **Social engineering targeting grid operators**: Social engineering attacks specifically target grid operators and utility corporations. Attackers may impersonate executives, technical support personnel, or government officials to manipulate employees into providing sensitive information or granting unauthorized access to critical systems (Tony Flick, 2010). Educating and training personnel to identify and mitigate social engineering risks is crucial, as these attacks exploit human vulnerabilities.

Blockchain exploitation: Blockchain technology promises to enhance the security and reliability of smart grids. However, attackers may exploit flaws in blockchain

implementations or smart contracts to obtain unauthorized access, tamper with transaction records, or disrupt the operation of blockchain-based systems. Integrity and resilience must be ensured for blockchain networks facilitating smart grid operations (Falahi et al., 2022; Gao et al., 2022; Kuzlu et al., 2020). **Data integrity attacks**: The smart grid heavily relies on accurate and reliable data for decision-making and control. Data integrity can be compromised by tampering with or manipulating data in transit or at rest. By compromising the integrity of grid data, adversaries can deceive operators, disrupt system operations, or contribute to operational failures or safety hazards (Gunduz and Das, 2020). **Interdependencies with other systems**: The smart grid is interdependent with a variety of other critical infrastructure systems, such as transportation, water supply, and telecommunications. Attackers may exploit the interdependencies between these systems to initiate coordinated attacks. Degrading telecommunications infrastructure, for instance, can impede grid monitoring and response capabilities, amplifying the effects of a cyberattack on the smart grid.

While cyberattacks dominate smart grid security discussions, the physical infrastructure can also be a target. Attackers may attempt to physically damage or eliminate essential grid components, such as substations, transmission lines, and control centers. Such assaults can result in prolonged outages, cascading failures, and significant monetary and societal repercussions. **Nation-state attacks:** Smart grids are vital components of the nation's infrastructure, making them potential targets for cyberattacks orchestrated by nation-states. Nation-state actors are capable of conducting sophisticated, long-term espionage, disruption, or subversion campaigns. These attacks may employ APTs, zero-day exploits, and complex attack vectors designed to exploit specific vulnerabilities and obtain strategic advantages.

Insider privilege abuse: In addition to insider threats, insiders with excessive privileges can abuse their access to critical systems. Privilege abuse can involve unauthorized configuration changes, bypassing security controls, or granting access to unauthorized individuals or external attackers. This poses a significant risk to the smart grid's security and operational integrity. **Insider-enabled physical attacks**: Insiders with knowledge of the smart grid's physical infrastructure and security systems can facilitate physical attacks by providing critical information to external threat actors. This collaboration between insiders and external adversaries can lead to targeted attacks on critical components, such as disabling physical security measures, compromising control systems, or damaging critical infrastructure.

Coordinated grid-scale attacks: Attackers may simultaneously and systematically target multiple components or regions of the smart grid. These grid-scale attacks can potentially overwhelm the system's defenses, resulting in pervasive disruptions or cascading failures. By exploiting vulnerabilities across multiple grid infrastructure layers, adversaries can maximize their impact and disrupt the grid's overall functionality. **Wireless network exploitation**: Wireless communication networks

play a crucial role in the smart grid, enabling data transmission, device control, and monitoring. Attackers may exploit vulnerabilities in wireless protocols or compromise wireless access points to gain unauthorized access to the grid's network. They can eavesdrop on communications (Voas, 2016), inject malicious commands, or disrupt wireless connectivity, leading to control system failures or unauthorized access to critical infrastructure.

Energy theft and fraud: Intelligent grids rely on precise measurement and invoicing systems to guarantee equitable energy distribution and billing. However, attackers may attempt to manipulate (Voas, 2016), smart meters, energy data, or invoicing systems to commit energy theft or fraud. This can lead to revenue losses for utility companies and compromise the grid's overall reliability (Tan et al., 2017). **Third-parties integration risks**: The smart grid ecosystem frequently integrates third-party applications, services, or devices. While this integration may provide benefits, it may also introduce security hazards. Attackers may target vulnerabilities in third-party systems to obtain unauthorized access to the smart grid infrastructure or exploit vulnerabilities at integration points, thereby compromising the grid's overall security posture.

Over-the-air attacks(OTA): Smart grid components, such as smart meters or grid sensors, often receive firmware updates or configuration changes OTA. Attackers may attempt to intercept OTA communications or tamper with the update process, injecting malicious firmware or commands into devices. These OTA can compromise the integrity of devices, leading to unauthorized control, data manipulation, or even physical damage to the grid infrastructure. **Cryptocurrency mining malware**: Cryptocurrency mining malware, also known as crypto-jacking, has become prevalent in various industries. Attackers can compromise devices within the smart grid infrastructure to mine cryptocurrencies, causing resource depletion, reduced performance, and increased energy consumption. This impacts the grid's operational efficiency, can strain the power supply, and lead to financial losses. **The emergence of quantum computing**: While still in its early stages, it poses opportunities and challenges for smart grid security. Quantum computers have the potential to break encryption algorithms commonly used to secure grid communications and data. As quantum computing advances, it becomes crucial to develop post-quantum encryption methods to ensure the long-term security of the smart grid.

Lack of standardized security practices: The smart grid ecosystem involves various stakeholders, including utility companies, manufacturers, vendors, and regulatory bodies. The lack of standardized security practices and protocols across these entities can create vulnerabilities and inconsistencies in security implementations (Vaos et al., 2018). Harmonizing security practices and establishing industry-wide standards can help ensure a more robust and resilient smart grid infrastructure.

Social media and open-source intelligence exploitation: Attackers can use in-

formation disseminated on social media platforms and other public sources (Tony Flick, 2010) to gather intelligence about the smart grid infrastructure. By analyzing publicly accessible data, adversaries can identify grid vulnerabilities, weaknesses, and potential targets, allowing them to plan and execute more effective cyberattacks. **IoT device vulnerabilities**: The proliferation of IoT devices in the smart infrastructure introduces a new class of vulnerabilities. IoT devices may have insufficient security features, obsolete firmware, or default credentials, making them vulnerable to compromise. Intruders may target these devices to obtain unauthorized access (Vaos et al., 2018) to the grid network, launch attacks, or as entry points for further infiltration.

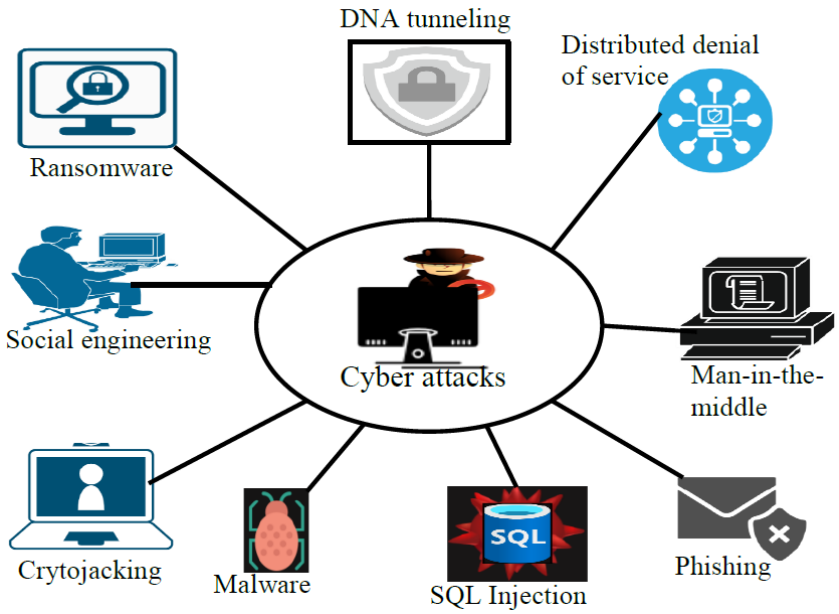


Figure 5: Different types of cyber-attacks.

Privacy breaches: Smart grid systems generate vast amounts of data about energy consumption, user behavior, and grid operations. Ensuring the privacy of this sensitive information is crucial. Attackers may target privacy vulnerabilities to gain unauthorized access to personal data or compromise the confidentiality (Vaos et al., 2018) of grid-related information. Privacy breaches can lead to public distrust, regulatory issues, and legal consequences. **Artificial intelligence-generated attacks**: Adversaries can leverage artificial intelligence technologies to automate and enhance attack capabilities. They can use artificial intelligence algorithms to generate sophisticated phishing emails, create targeted spear-phishing campaigns, or even develop artificial intelligence-driven malware that can adapt and evolve to bypass traditional security defenses. Artificial intelligence-generated attacks pose signifi-

cant challenges in detection and mitigation.

5G network exploitation: As 5G networks are increasingly deployed, smart grids are anticipated to exploit this technology's benefits, such as decreased latency and increased bandwidth. Nevertheless, the implementation of 5G networks introduces potential security hazards. Attackers may exploit 5G infrastructure vulnerabilities, such as signaling protocols or network slicing, to disrupt grid communications or obtain unauthorized access to the smart grid. Implementing intricate security layers within the 5G network could potentially result in elevated latency levels. Given that numerous control and protection applications within smart grids are exceptionally latency-sensitive, this consideration becomes paramount. **Virtualization and cloud-based attacks**: Virtualization and cloud computing are indispensable to modernizing the smart grid. However, these technologies introduce their security risks. In order to obtain unauthorized access to grid systems, manipulate data, or disrupt vital operations, attackers may exploit vulnerabilities in virtualized environments or compromise cloud-based services (Vaos et al., 2018).

Insider threats from external contractors: The smart grid ecosystem frequently includes transient grid infrastructure access for external contractors. These contractors may have access to privileged systems and data, making them potential insider threats. Organizations must implement robust access controls (Eken, 2013), monitoring mechanisms, and rigorous verification procedures to mitigate the risks associated with external contractors. **Infrastructure-as-a-service (IaaS) attacks**: Smart grid operators may host critical applications or infrastructure components with IaaS providers. To obtain unauthorized access to the smart grid infrastructure, attackers can target vulnerabilities in IaaS platforms or exploit misconfigurations. IaaS environments must implement robust security measures and undergo regular security assessments. **Physical security system vulnerabilities**: Physical security systems, such as surveillance cameras, access control systems, and IDS, are critical for protecting the smart grid's physical infrastructure. Attackers may target vulnerabilities in these systems to bypass or disable them, facilitating unauthorized physical access to critical grid components.

As technology evolves and cybercriminals become more sophisticated, new cyber threats and attack vectors will emerge. To remain ahead of cyber adversaries and ensure the secure and dependable operation of the smart grid, ongoing research, collaboration, and the continuous improvement of security measures are essential. The evolving threat landscape necessitates continuous monitoring, evaluation, and enhancement of the smart grid's security measures. Implementing a defense-in-depth strategy consisting of technical controls, security awareness training, incident response plans, and regulatory frameworks are crucial for mitigating the risks posed by these emergent cyber-security threats and attacks. Emerging cyber threats and cyberattacks highlight the need for a comprehensive, multi-layered smart grid security strategy. To ensure the resilience and dependability of the smart grid, it is nec-

essary to secure not only the cyber-infrastructure but also physical security, training, awareness programs, incident response capabilities, regulatory frameworks, and collaborations between industry stakeholders, government agencies, and cybersecurity experts.

3 MACHINE LEARNING AND CYBER-SECURITY IN SMART GRID

3.1 Machine learning

Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and models that enable computers to learn from data and make predictions or in general modeling without being explicitly programmed. It involves studying statistical techniques, linear algebra, and computational models that automatically learn patterns and relationships from data, allowing machines to improve their performance over time (Batta, 2018). It has gained significant attention in cybersecurity due to its ability to analyze large-scale data, detect anomalies, and identify patterns that may indicate malicious activities.

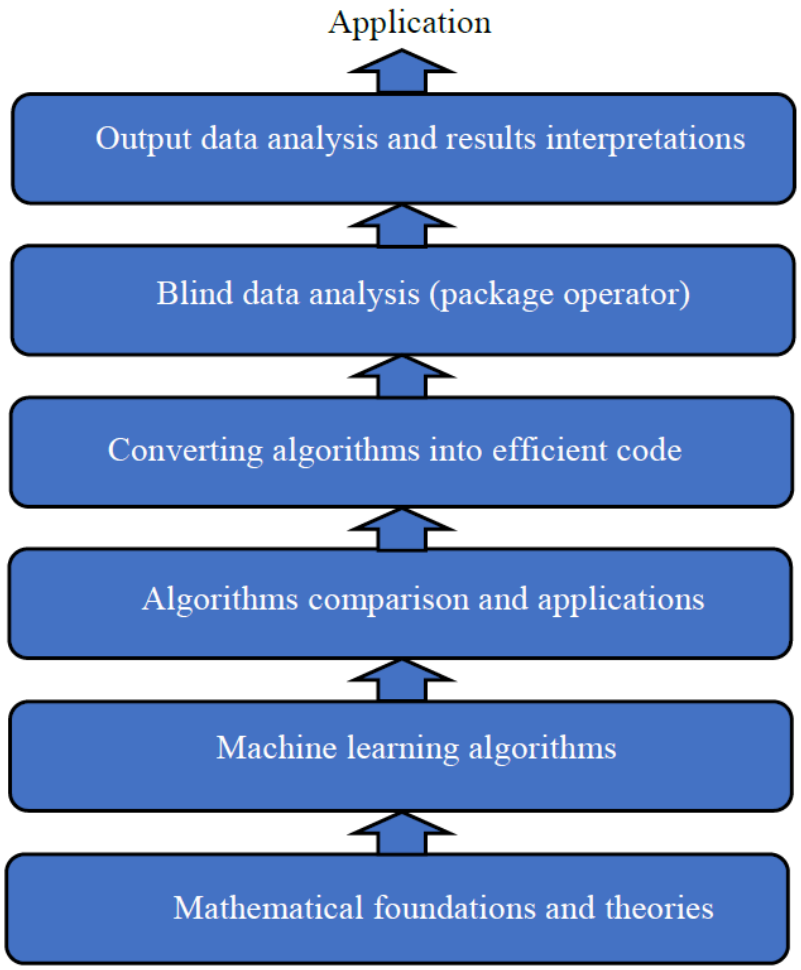


Figure 6: Machine learning layers.

In machine learning, the learning process starts with training data consisting of input variables (features or attribute) and corresponding output variables (labels or target values). The algorithm analyses the training data to identify patterns, dependencies, and statistical relationships. The goal is to create a model that can generalize from the training data and make accurate predictions or decisions on unseen or future data. There are various machine learning algorithms, such as supervised, unsupervised, semi-supervised, and reinforcement learning (Teixeira et al., 2018).

- Supervised learning: This type of learning involves training a model with labeled data, where the input features and corresponding output labels are provided. The model learns to map the input features to the output labels, enabling it to make predictions on new unseen data (Alkuwari et al., 2022; Sarker, 2022a).
- Unsupervised learning: Unsupervised learning deals with unlabeled data, where only the input features are provided. The algorithm learns to discover hidden patterns, structures, or clusters within the data without explicit guidance. It helps find insights and understand the data's structure (Alkuwari et al., 2022; Sarker, 2022a). In essence, the task involves discovering the optimal multivariate probability density function characterized by parameters that accurately capture the underlying data.
- Semi-supervised learning: Semi-supervised learning combines elements of both supervised and unsupervised learning. It leverages a small amount of labeled data and a more considerable amount of unlabeled data to create models that can make predictions or decisions (Alkuwari et al., 2022; Azad et al., 2019).
- Reinforcement learning involves training an agent to interact with an environment and learn through trial and error. The agent receives feedback in the form of rewards or penalties based on its actions, allowing it to learn optimal strategies or policies to maximize long-term rewards (Alkuwari et al., 2022; Sarker, 2022a; Syrmakesis et al., 2022)

Machine learning algorithms utilize various techniques and methods such as regression, classification, clustering, dimensionality reduction, neural networks, decision trees, support vector machines (SVM), and more. These algorithms can be applied to a wide range of domains, including image recognition, speech recognition, natural language processing, recommendation systems, fraud detection, autonomous vehicles, healthcare, finance, and many others (Sarker et al., 2021). In summary, machine learning enables computers to learn from data and make predictions or choices without explicit programming. It makes use of algorithms and models that examine patterns and relationships in the data, enabling machines to perform better and offer insightful analyses and predictions across a range of industries.

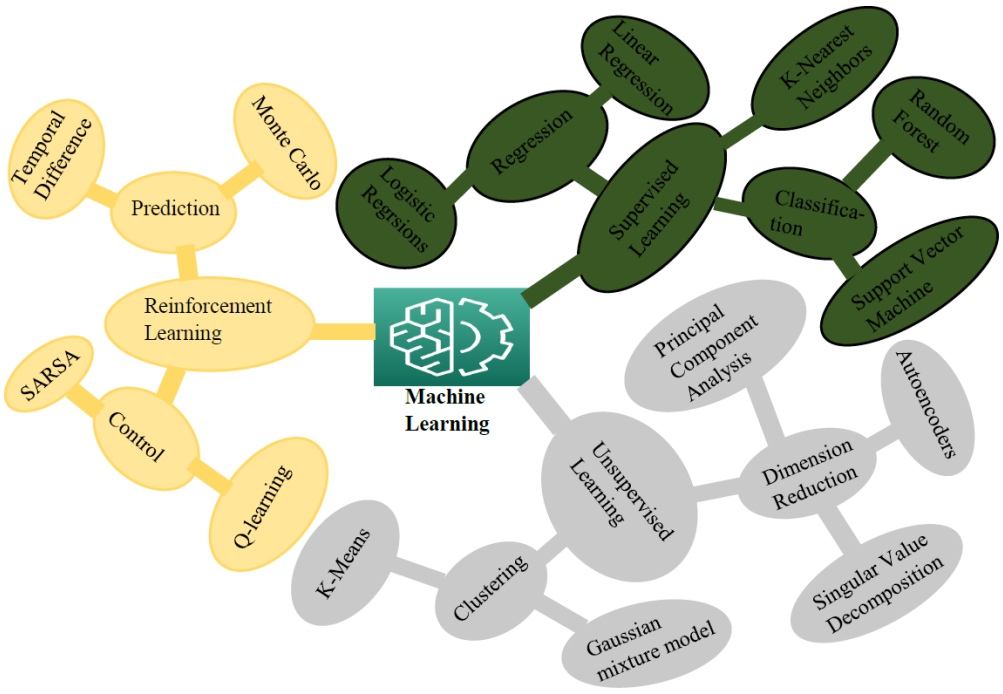


Figure 7: A categorization of major machine learning techniques with relevant examples.

3.2 Machine learning algorithms for cyber-security in smart grid

Due to the smart grid system's complexity, magnitude, and interconnectedness, it is extremely susceptible to cyber-attacks (Dogaru & Dumitrache, 2019; Y. Li et al., 2022); machine learning algorithms are vital tools in identifying and mitigating such attacks. Smart grids face threats such as data breaches, unauthorized access, malware attacks, and denial-of-service (DoS) attacks. These threats can have severe consequences, including disruption of electricity supply, compromising customer privacy, and even physical damage to the grid infrastructure. In smart grid cyber-security, machine learning algorithms must be used to safeguard the infrastructure from such threats. Machine learning algorithms make real-time analysis and detection of aberrant activity feasible, allowing for the quick implementation of preventative actions. Cyber dangers are evolving along with technology; thus, it is essential to create machine learning algorithms to strengthen smart grids' security. Some essential machine-learning techniques that can help identify and mitigate potential threats are:

Intrusion detection systems (IDS): Machine learning algorithms are extensively used in IDS to detect and prevent cyber-attacks in smart grids (Siraj, 2014). These algorithms analyze network traffic, system logs, and other relevant data to identify patterns and anomalies associated with malicious activities. They can learn from

historical data to identify new and emerging attack patterns, enabling the system to detect and respond to previously unknown threats (Nguyen & Reddi, 2021; Prabhakar et al., 2022). These models are then deployed to continuously monitor the smart grid infrastructure and detect any suspicious or abnormal behavior that deviates from expected patterns. Examples of machine learning algorithms used in IDS include decision trees, SVM, random forests, and deep learning models like convolutional neural networks (CNN) and recurrent neural networks (RNN). It is worth noting that machine learning algorithms used in IDS should be regularly updated and trained with the latest data to maintain their effectiveness.

Besides, the performance of these algorithms depends on the quality and representativeness of the training data, as well as the proper selection and tuning of algorithm parameters. So, a comprehensive and proactive approach to smart grid security should incorporate machine learning-based IDS and other security measures such as network segmentation, access control, encryption, and regular security audits. Malware detection: Smart grids are vulnerable to malware attacks that can disrupt operations and compromise the entire system's security. Machine learning algorithms can help in the early detection of malware by analyzing system behavior and identifying malicious code or activities. These algorithms can learn from large datasets of known malware samples and detect variations or new types of malware (Liang et al., 2018). Techniques such as static and dynamic analysis, feature extraction, and clustering algorithms are employed for effective malware detection in smart grids.

In the context of smart grids, machine learning algorithms can be trained using large datasets of known malware samples. These datasets may include various types of malware, such as viruses, worms, Trojans, and ransomware, that have been previously identified and analyzed. By learning from these datasets, the algorithms can develop models that can effectively detect variations or new types of malware that have not been previously encountered. Feature extraction is an important step in the process of malware detection. Machine learning algorithms analyze system data and extract relevant features or characteristics that can distinguish between normal and malicious behavior. These features include network traffic patterns, system resource usage, communication protocols, and data access patterns. By extracting meaningful features, the algorithms can identify behavioral patterns associated with malware activities and flag any suspicious behavior for further investigation.

Clustering algorithms play a crucial role in organizing and categorizing malware samples. These algorithms group similar malware samples based on their characteristics, enabling security analysts to understand the commonalities and variations among different types of malware. By clustering malware samples, machine learning algorithms can effectively classify and identify new or unknown malware based on their similarity to previously encountered samples. This enables the early detection and classification of emerging malware threats, even if they have not been

previously identified.

Applying machine learning algorithms in smart grid security provides an intelligent, automated approach to detecting malware. By continuously analyzing system behavior, these algorithms can identify potential threats, alert security personnel, and facilitate prompt response and mitigation actions. However, while machine learning algorithms can significantly improve malware detection in smart grids, a comprehensive security strategy should include other measures such as robust network architecture, secure communication protocols, and regular software updates to ensure the system's overall resilience.

Anomaly detection is crucial for identifying unusual behavior or deviations from normal patterns in smart grids. Machine learning algorithms can analyze vast amounts of sensor data, network logs, and system performance metrics to establish normal behavior profiles and identify deviations that may indicate potential security breaches. The machine learning algorithms employed for anomaly detection in smart grids often fall under the category of unsupervised learning. Unlike supervised learning, unsupervised learning algorithms do not require labeled data for training. Instead, they focus on discovering patterns and structures in the data, including deviations from normal behavior (Shampa et al., 2023). Unsupervised learning algorithms like clustering, autoencoders, and Gaussian mixture models are commonly used for anomaly detection in smart grids (Xiufeng Liu, 2016). By continuously monitoring and analyzing system behavior, these algorithms can detect intrusions, unauthorized access attempts, or abnormal activities in real-time.

Autoencoders are another powerful technique for anomaly detection. Autoencoders are highly non-linear black box like neural networks architectures that are trained to reconstruct the input data from a compressed representation called the latent space. During the training phase, the autoencoder learns to encode and decode normal data accurately. When presented with anomalous data, the reconstruction error is typically higher, indicating a deviation from the expected patterns. Autoencoders can detect real-time anomalies by setting a threshold for the reconstruction error. Gaussian mixture models (GMMs) are probabilistic models that assume the data is generated from a mixture of Gaussian distributions. GMMs can capture the statistical properties of the normal data distribution and estimate the likelihood of new data points belonging to the same distribution. Data points with low likelihood values are considered anomalous. GMMs are particularly useful for detecting anomalies in multi-dimensional data where complex distributions can characterize normal behavior (de Souza et al., 2022).

Machine learning algorithms for anomaly detection in smart grids can detect various security breaches by continuously monitoring and analyzing system behavior. They can identify intrusions, unauthorized access attempts, system malfunctions, abnormal energy consumption, or other activities deviating from established nor-

mal patterns. Real-time anomaly detection enables rapid response and mitigation, minimizing the potential impact of cyber-attacks or operational failures. Predictive maintenance: Machine learning algorithms can also contribute to the security of smart grids through predictive maintenance. These algorithms can predict potential failures or vulnerabilities (Chicco et al., 2020) in the grid infrastructure by analyzing historical sensor data, system logs, and maintenance records. Predictive maintenance models can help identify potential security weaknesses and proactively address them before attackers exploit them. Timely maintenance and security updates can significantly reduce the risk of cyber-attacks and system failures.

Predictive maintenance plays a crucial role in ensuring the security and reliability of smart grid infrastructure. By leveraging machine learning algorithms, historical data, and advanced analytics, predictive maintenance can identify potential failures, vulnerabilities, or deteriorating conditions in the grid components. This proactive approach enables operators to address security weaknesses and mitigate risks before they lead to cyber-attacks or system failures (T. Li et al., 2019). Machine learning algorithms used in predictive maintenance analyze various data sources, including historical sensor data, system logs, maintenance records, and external data such as weather conditions or grid load patterns. These algorithms can identify patterns, correlations, and anomalies in the data to develop models that predict the future behavior of grid components. The models can provide insights into the strength and performance of smart grid infrastructure. They can predict the likelihood and timing of component failures, detect abnormal behaviors, and identify potential security vulnerabilities that attackers may exploit. By leveraging historical data, the algorithms can learn the patterns associated with failures or security incidents and use this knowledge to make accurate predictions.

Integrating security-related data into predictive maintenance models allows for identifying potential vulnerabilities in the grid infrastructure. For example, machine learning algorithms can detect unusual access patterns, unauthorized activities, or signs of cyber-attacks by analyzing system logs and network traffic data. The models can highlight areas where security updates or patches are needed to address vulnerabilities and prevent potential breaches. Timely maintenance and security updates based on the predictions of machine learning algorithms can significantly reduce the risk of cyber-attacks and system failures. By proactively addressing potential failures or vulnerabilities, operators can mitigate the impact on the smart grid's security and performance. This approach not only enhances the security of the grid but also improves the overall efficiency and reliability of the system.

Additionally, predictive maintenance can optimize maintenance schedules and resource allocation. By predicting when specific components are likely to fail or require maintenance, operators can plan maintenance activities more effectively. This approach minimizes the downtime and disruption caused by unexpected failures, reduces maintenance costs, and maximizes the lifespan of grid assets. To

ensure the effectiveness of predictive maintenance algorithms, continuously updating and retraining the models with new data is essential. As the smart grid evolves, new technologies are introduced, and operational conditions change, the algorithms must adapt to capture the changing patterns and behaviors. Integrating real-time data streams into the predictive maintenance models can further enhance their accuracy and responsiveness to evolving security threats. In conclusion, predictive maintenance powered by machine learning algorithms offers a proactive and intelligent approach to ensuring the security and reliability of smart grid infrastructure. By analyzing historical data and identifying potential failures, vulnerabilities, and security weaknesses, operators can take timely actions to prevent cyber-attacks, reduce downtime, optimize maintenance activities, and enhance the overall performance of the smart grid.

User encryption and authorization: Ensuring the identity and authorization of users accessing the smart grid infrastructure is vital for maintaining security. Machine learning algorithms can be utilized for user authentication and access control by analyzing user behavior patterns, biometric data, and historical usage patterns (Wang and Lu, 2013). These algorithms can identify suspicious login attempts, detect unauthorized access, and flag potential security threats in real time. Advanced techniques like deep learning-based facial recognition and voice authentication systems are gaining prominence for user authentication in smart grids. The facial recognition systems utilize neural networks to analyze facial features and match them against a database of known users. By capturing and analyzing facial biometric data during the login process, deep learning algorithms can accurately verify the identity of users, making it difficult for unauthorized individuals to gain access.

Another advanced technique used for user authentication in smart grids is voice authentication, which analyzes voice patterns, pitch, intonation, and other acoustic characteristics to verify the identity of users. By comparing the voice samples provided during the authentication process with pre-registered voice templates, the algorithms can determine whether or not the user is legitimate. Machine learning algorithms can also be utilized for continuous authentication, where user behavior is monitored throughout a session to detect suspicious activities. These algorithms can analyze user actions, mouse movements, keystrokes, and other behavioral patterns to ensure the ongoing legitimacy of user access. If the algorithms detect unexpected changes in behavior, such as erratic mouse movements or a sudden change in typing patterns, they can prompt additional authentication steps or terminate the session. It is crucial to highlight that privacy concerns must be carefully addressed (Al Ameen et al., 2012) when implementing machine learning-based user authentication systems. Proper data anonymization, encryption (Diamantoulakis et al., 2015), and secure storage of biometric data are essential to protect user privacy. Also, transparency and user consent should be ensured, and appropriate legal and ethical guidelines should be followed when collecting and utilizing biometric data.

Secure communication networks: Securing communication networks within a smart grid is essential to protect against various types of attacks and maintain data confidentiality, integrity, and availability. Machine learning algorithms offer valuable techniques for enhancing the security of communication channels by encrypting data, authenticating communication nodes, detecting tampering attempts, and optimizing secure communication protocols. Encryption is a fundamental method for protecting data during transmission. Machine learning algorithms can assist in developing robust encryption algorithms and optimizing encryption parameters based on various factors such as data sensitivity, network conditions, and security requirements. These algorithms can analyze historical data and patterns to identify potential vulnerabilities in encryption schemes and contribute to the development of stronger encryption methods. Authentication of communication nodes ensures that only authorized entities can participate (Bhattarai et al., 2019) in the smart grid network. Machine learning algorithms can play a role in authentication processes by analyzing authentication data, user behavior, and historical usage patterns. These algorithms can learn from data and develop models that accurately verify the identity of communication nodes, preventing unauthorized access or impersonation.

The smart grid can establish a more reliable and secure communication infrastructure by incorporating machine learning into authentication mechanisms. Analyzing network traffic, monitoring data authentication, and identifying patterns indicative of tampering or intrusion attempts are crucial for securing communication networks in smart grids. Historical data can be leveraged to establish normal behavior profiles, enabling the detection of deviations or anomalies in real time. Timely alerts are raised when suspicious activities are detected, allowing for proactive detection and mitigation of tampering attempts. This approach enhances the overall security of the communication network.

Reinforcement learning algorithms can optimize secure communication protocols by learning from environmental interactions and adapting to emerging threats. These algorithms can utilize feedback from the system to improve the efficiency and effectiveness of communication protocols, dynamically adjusting parameters and configurations to mitigate vulnerabilities or adapt to changing attack techniques. Reinforcement learning enables the development of adaptive and resilient communication protocols that can respond to new threats in real time. Machine learning algorithms can also contribute to anomaly detection in communication networks. By analyzing network traffic data, system logs, and performance metrics, these algorithms can identify abnormal communication patterns that may indicate potential security breaches or attacks. Anomalies in network traffic, such as unexpected spikes in data volume or unusual communication patterns, can be flagged as potential threats, prompting further investigation and response.

It is important to note that the security of communication networks requires a multilayered approach that combines machine learning techniques with other security

measures such as firewalls, intrusion detection systems, and secure protocols. Machine learning algorithms complement and enhance existing security mechanisms, providing intelligent and adaptive capabilities to protect communication channels within the smart grid. In summary, machine learning algorithms offer valuable contributions to securing communication networks in smart grids. They can assist in encryption, authentication, tampering detection, optimization of communication protocols, anomaly detection, and intrusion detection. By leveraging the power of machine learning, smart grids can establish robust and resilient communication infrastructures that protect against eavesdropping, tampering, and replay attacks, ensuring the secure and reliable transmission of data within the grid.

Threat intelligence and risk assessment: Threat intelligence and risk assessment are crucial aspects of maintaining the security and resilience of smart grid systems. Machine learning algorithms can collect, analyze, and interpret vast amounts of security-related data to provide valuable insights into potential risks and vulnerabilities. By mining and processing data from diverse sources, these algorithms enable security teams to prioritize threats, assess their impact, and develop effective mitigation strategies. Once the data is collected, machine learning algorithms can analyze and process it to identify patterns, correlations, and emerging trends. These algorithms can learn from historical data and discover relationships between different threat indicators, helping to identify potential attack vectors or vulnerabilities specific to the smart grid environment. By detecting patterns associated with known attacks or indicators of compromise, the algorithms can provide early warnings and insights into potential risks.

Risk assessment is a critical component of security management. Machine learning algorithms can contribute to risk assessment by evaluating the impact and likelihood of various threats (Lamba et al., 2019). By considering factors such as the severity of vulnerabilities, the possibility of exploitation, and the potential impact (Mohammed et al., 2023) on the smart grid infrastructure, these algorithms can help prioritize risks and guide the allocation of resources for mitigation efforts. They can also assist in developing risk-scoring models. These models assign scores or ratings to different threats based on their potential impact and likelihood. By integrating various data sources and applying advanced analytics techniques, the algorithms can quantitatively assess risks, enabling security teams to prioritize and address the most critical threats first.

Furthermore, the algorithms can aid in the development of effective mitigation strategies. These algorithms can learn from past experiences and recommend suitable countermeasures for specific threats by analyzing historical data on successful and unsuccessful mitigation efforts. They can identify patterns of successful responses and provide insights into the most effective mitigation techniques for different types of attacks. The continuous learning and adaptation capabilities of machine learning algorithms make them well-suited for the dynamic nature of the

threat landscape. These algorithms can adapt to changing attack techniques, evolving vulnerabilities, and emerging risks. By continuously analyzing new data and learning from ongoing incidents, the algorithms can update their models and provide timely and relevant threat intelligence to security teams.

3.2.1 Decision tree

Decision trees are a popular and widely used machine learning algorithm that can be applied to a variety of problem domains, including cyber-security in the smart grid. Decision trees are supervised machine learning algorithms that learn from labeled training data to make predictions or decisions (Charbuty & Abdulazeez, 2021). They construct a tree-like model where each internal node represents a feature or attribute, each branch represents a decision rule based on that attribute, and each leaf/terminal node represents a class label or an outcome (Pandey & Kumar Sharma, 2013).

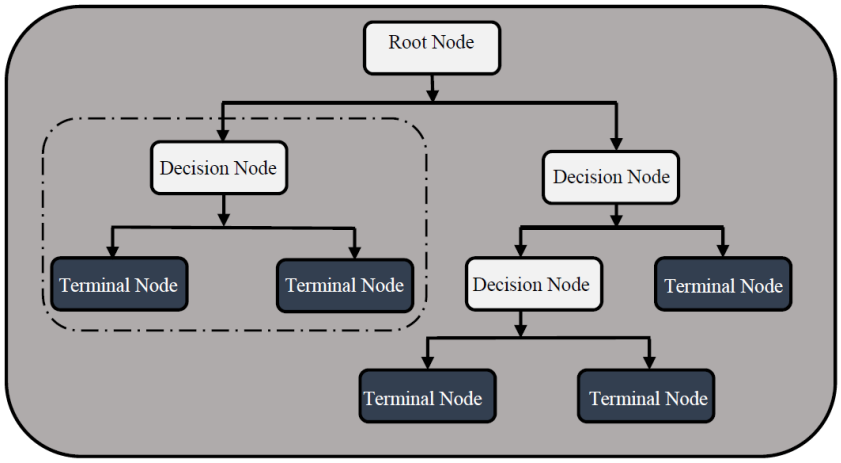


Figure 8: Pictorial representation of a decision tree structure.

The construction of a decision tree involves recursively partitioning the training data based on different features, selecting the best feature that provides the most significant information gain or impurity reduction at each step. The goal is to create partitions that are as pure as possible, meaning they contain mostly instances of a single class. This process results in a tree structure that can be used for classification or regression tasks. It works by recursively partitioning the training data based on different features, selecting the best feature that provides the most significant information gain or impurity reduction at each step. The overview of how the decision tree algorithm works are:

Data preparation: The decision tree algorithm begins by preparing the training data. Each data instance contains features or attributes and a corresponding class label or outcome. If the data contains categorical features, they may need to be encoded into numerical values to be processed by the algorithm. Attribute Selection: The algorithm selects the best attribute or feature that will serve as the root of the decision tree. The attribute selection is based on a measure of impurity or information gain, which evaluates how well an attribute can split the data and improve the classification or regression performance. Common impurity measures include Gini impurity and information gain (based on entropy).

Partitioning: Once the root attribute is selected, the data is partitioned into subsets based on the attribute's possible values. Each subset corresponds to a branch of the decision tree emanating from the root node. The partitioning process divides the data based on the attribute's values, assigning each instance to the appropriate branch.
Recursion: The algorithm recursively repeats attribute selection and partitioning for each subset or branch created in the previous step. It selects the best attribute for each subset based on the impurity or information gain criterion. This process continues until a stopping condition is met, such as reaching a predefined maximum depth or having a minimum number of instances in a leaf node. Leaf node creation: At each step, if a stopping condition is met, the algorithm creates a leaf node and assigns a class label or outcome based on the majority class of the instances in that subset. For regression tasks, the leaf node may contain a predicted numerical value based on the average or median of the instances in that subset.

Tree pruning: After the decision tree is constructed, pruning techniques may be applied to reduce overfitting. Pruning involves removing branches or nodes that do not contribute significantly to improving the overall performance on unseen data. This helps the decision tree generalize better and avoid memorizing the training examples. Prediction: Once the decision tree is constructed, it can be used to make predictions on unseen data. For a given instance, the algorithm follows the decision rules in the tree, traversing from the root to a leaf node based on the attribute values of the instance. The predicted class label or outcome is the value associated with the leaf node reached. Some key characteristics and advantages of decision trees are:

1. Interpretability: Decision trees offer high interpretability, as humans can easily visualize and understand the resulting tree structure. Each decision rule and branching point in the tree can be interpreted as a set of conditions or criteria that lead to a particular outcome. This interpretability makes decision trees valuable for gaining insights into the decision-making process.

2. Handling both categorical and numerical data: Decision trees can handle both categorical and numerical features, making them versatile for various types of data in the context of smart grids. The algorithm can handle features with

discrete categories or continuous values and automatically determines the best splitting points during the tree construction process.

3. Feature importance and selection: Decision trees provide a measure of feature importance by evaluating the contribution of each feature to the overall tree structure. Features higher up in the tree and resulting in significant impurity reduction or information gain are considered more important. This information can be utilized for feature selection and identifying the most influential factors in cyber-security analysis.

4. Non-linear relationships: Decision trees are capable of capturing non-linear relationships between features and class labels. By recursively partitioning the data, decision trees can effectively model complex decision boundaries that separate different classes.

5. Robustness to outliers and noise: Decision trees are relatively robust to outliers and noisy data points. The tree construction process is guided by impurity reduction, making it less sensitive to individual data points that deviate from the majority. However, preprocessing and cleaning the data is still important to minimize the impact of outliers and noise.

6. Scalability and efficiency: Decision tree algorithms, such as the popular classification and regression trees (CART) algorithm, have efficient implementation techniques and can handle large datasets with reasonable computational resources. The prediction time complexity of decision trees is logarithmic concerning the number of instances in the tree, making them suitable for real-time applications in smart grids.

While decision trees offer several advantages, they also have some limitations: Overfitting: Decision trees are prone to overfitting, particularly when the tree becomes too complex and captures noise or specific characteristics of the training data. Overfitting occurs when the tree memorizes the training examples instead of generalizing well to unseen data. Techniques like pruning, maximum depth setting, or ensemble methods like random forests can help alleviate overfitting. Lack of robustness to small changes: Decision trees are sensitive to small changes in the training data, which can lead to different tree structures. This sensitivity can make decision trees less stable compared to other algorithms. Ensemble methods like random forests can address this issue by combining multiple decision trees to make more robust predictions. Bias towards features with more levels: Decision trees favor features with more levels or categories during tree construction. This bias can lead to uneven splits and potentially overlook important features with fewer levels.

3.2.1.1 Classification and regression trees

CARTs are powerful machine learning algorithms that can be used for both classification and regression tasks. CART builds a binary tree structure by recursively

partitioning the input space based on feature values, resulting in a series of decision rules (Taghavinejad et al., 2020). These decision rules are used to classify or predict the target variable for new data instances. As a result, because it enables feature selection, it has emerged as the perfect tool for data analysis. Regression trees are frequently used when the goal variable has a numerical form, and the mean of the responses in a given region will serve as the terminal node's mean value. The mean value will be used to predict any new or unusual data or observations. When the targeted variable is categorical, classification techniques are used, and the response mode throughout that region will represent the value found at the terminal node. Thus, any new data or observation within that category will have its prediction made based on the modal value. CART algorithms have several key characteristics and benefits:

Tree structure: CART algorithms create a tree-like structure where each internal node represents a feature test or decision rule, and each leaf node represents a class label (in classification) or a predicted value (in regression). The tree structure provides an intuitive representation of the decision-making process.

Recursive partitioning: CART employs a top-down recursive partitioning approach. It begins with the entire dataset and recursively splits it into subsets based on the values of different features. The splitting process continues until a stopping criterion is met, such as reaching a maximum tree depth, achieving a minimum number of samples per leaf, or when further splitting does not improve the model's performance significantly.

Feature selection: CART determines the optimal feature and splitting point at each internal node based on criteria that maximize the separation between classes or minimize the variance in the target variable. Gini impurity and information gain are commonly used criteria for classification, while mean squared error and mean absolute error are used for regression.

Handling categorical and numeric features: CART algorithms can handle both categorical and numeric features. The algorithm partitions the data based on equality or inequality with specific categories for categorical features. The algorithm selects a threshold for numeric features to split the data into two groups based on whether the feature value is greater than or equal to the threshold.

Model interpretability: CART models are highly interpretable. The decision rules represented by the tree structure provide clear explanations for the classification or regression outcomes. It is easy to understand how each feature contributes to the final prediction and trace the decision path through the tree.

Handling missing values and outliers: CART algorithms can assign them to the most probable class or use surrogate splits. They are also robust to outliers since the splitting criteria are based on relative measures, rather than absolute values.

Ensemble methods: CART can be combined with ensemble methods, such as random forests or gradient boosting, to improve predictive performance. These ensemble methods create multiple CART models and combine their predictions to achieve

better generalization and reduce overfitting.
Zooming on to some various applications of the CART algorithms:

Classification: CART is widely used for classification tasks, such as email spam detection, fraud detection, disease diagnosis, and sentiment analysis. It can handle both binary and multi-class classification problems.
Regression: CART can also be applied to regression problems, such as predicting house prices, demand forecasting, or energy consumption prediction. It builds a regression tree to estimate the continuous target variable based on the input features.
Feature selection: CART can be used to select relevant features by measuring their importance in the tree construction process. Features that contribute significantly to the tree structure are important and can be used for subsequent analysis.
Anomaly detection: CART can be utilized for anomaly detection by constructing a classification tree and identifying instances that deviate significantly from the normal class. In conclusion, CARTs are versatile machine learning algorithms that can handle both classification and regression tasks. They provide interpretable models, handle missing values and outliers, and can be combined with ensemble methods. CART algorithms have a wide range of applications in various domains, making them valuable tools in data analysis and decision-making processes.

3.2.1.2 Entropy

Entropy is a concept from information theory that measures the impurity or disorder in a data set. In the context of machine learning, entropy is commonly used as a measure of impurity or disorder in a dataset. In the context of decision tree algorithms like CART, entropy is used as a criterion to determine the quality of splits during the tree-building process. In classification tasks, entropy is used to quantify the uncertainty or randomness associated with the distribution of class labels in a dataset. The entropy of a node represents the impurity of that node. A node with low entropy means the class labels are relatively pure, while a node with high entropy indicates a more mixed distribution of class labels. The entropy of node N is calculated using the following formula (Safavian and Landgrebe, 1991):

$$E(N) = -\sum_{i=1}^{c} P_i \times log_2(P_i) \tag{1}$$

where p_i is the proportion of instances belonging to class i in node N and c denotes the number of classes (Kurniabudi et al., 2020). The summation goes over all the classes in the dataset. The entropy value ranges from 0 to $log_2(C)$ of the number of classes in the dataset. A value of 0 indicates a node with pure class labels (all instances belong to the same class), while a value of 1 indicates a node with an equal distribution of class labels (maximum impurity). If you have a binary classification

problem (two classes), then the maximum entropy value is 1. However, if there are more than two classes, the maximum entropy value can be higher. In general, the maximum entropy value for a dataset with C classes is $log_2(C)$.

In the context of CART, entropy is used to evaluate the quality of a split. When deciding which feature to use for splitting a node, CART considers the reduction in entropy achieved by the split. The idea is to select the feature and threshold resulting in the largest entropy decrease among the child nodes. The split that minimizes entropy is considered the most informative and provides the greatest separation of classes. The reduction in entropy, often referred to as information gain, is calculated by subtracting the weighted average of the child nodes entropies from the parent node's entropy. The information gain quantifies the amount of information gained by splitting the data based on a particular feature. CART uses information gain (or Gini impurity, another node impurity measure) as the splitting criterion to construct decision trees. The algorithm recursively applies this criterion to select the best splits and build a tree until a stopping criterion is met. By using entropy as a measure of impurity and information gain as the splitting criterion, CART aims to create decision trees that effectively separate classes and make accurate predictions. The nodes with lower entropy represent more homogeneous subsets of data, allowing for better classification accuracy and interpretability of the resulting tree model.

3.2.1.3 Information gain

Information gain is a measure used in decision tree algorithms, such as CART, to evaluate the usefulness of a feature for splitting a node and building a decision tree. It quantifies the amount of information gained by splitting the data based on a particular feature. The amount of information that a node provides is measured by information gain. It evaluates how successfully a feature categorizes the class, where the maximum information-providing node is chosen. In other words, the information gain determines the entropy's decrease by quantifying the amount of information gained after the data split. The information gain is calculated by comparing the entropy (or Gini impurity) of the parent node with the weighted average of the entropies (or Gini impurities) of the resulting child nodes after the split. The idea is to select the feature and threshold that maximize the information gain, which indicates the most informative split. The information gain can be computed using the following formula:

$$Gain = E_{parent} - \sum [weighted\ average\ of\ child\ entropies] \tag{2}$$

where E_{parent} is the entropy of the parent node or the Gini impurity of the parent

node before the split. $\sum$ represents the summation of all child nodes resulting from the split. The weighted average of child entropies or child Gini impurities is the average entropy or Gini impurity of each child node, weighted by the proportion of instances in that child node.

A more significant information gain implies a better split since it shows that the split successfully divides the classes and lessens uncertainty or impurity in the resulting child nodes. A feature with a larger information gain is seen as more informative and offers more class separation. After assessing the information gain for each feature, the CART algorithm chooses the feature with the largest information gain as the splitting criterion for the current node. Until a stopping requirement is satisfied, such as reaching a maximum tree depth or a minimum number of instances per leaf, this process is repeated iteratively for each child node. By maximizing information gain, CART aims to create decision trees that efficiently partition the data based on the most informative features, resulting in nodes with lower entropy or Gini impurity and improved classification accuracy. It is important to note that while information gain is a commonly used criterion, other measures such as gain ratio and Gini index can also be used in decision tree algorithms to assess the quality of splits. The choice of criterion depends on the specific algorithm and problem at hand.

3.2.1.4 Gini index

The decision tree methods employed in CART, notably the Gini index, measure diversity or impurity. Based on the distribution of class labels in that node, it calculates the likelihood of incorrectly categorizing a randomly selected instance from that node. The Gini index is used to assess a node's impurity in terms of the distribution of class labels in the context of classification tasks. A node with a low Gini index is considered relatively pure, meaning that most of its instances are members of one class. On the other hand, a node with a high Gini index suggests a more erratic distribution of class labels, which denotes increased impurity. The Gini index of a node N is calculated using the following formula:

$$Gini(N) = -\sum_{i=1}^{c} P_i \times (1 - P_i) \tag{3}$$

where p_i is the proportion of instances belonging to class i in node N. The summation is performed over all unique class labels in node N. The Gini index ranges from 0 to 1, where a value of 0 represents a node with pure class labels (all instances belong to the same class), and a value of 1 represents a node with an equal distribution of class labels (maximum impurity).

In the context of CART, the Gini index is used as a criterion to evaluate the quality of splits during the tree-building process. When deciding which feature to use for splitting a node, CART considers the reduction in the Gini index achieved by the split. The idea is to select the feature and threshold that result in the largest decrease in the Gini index among the child nodes. The split that minimizes the Gini index is considered the most informative and provides the greatest separation of classes. Similar to information gain, CART aims to create decision trees that effectively separate classes by selecting splits that minimize the Gini index. The nodes with lower Gini index represent more homogeneous subsets of data, allowing for better classification accuracy and interpretability of the resulting tree model. It is important to note that information gain and the Gini index are two commonly used criteria for evaluating splits in decision tree algorithms. Both measures similarly assess the quality of splits and select informative features for building decision trees. The choice between information gain and the Gini index depends on the specific algorithm and problem context.

3.2.2 Support vector machine

SVM is a powerful and widely used machine learning algorithm for both classification and regression tasks. It effectively solves complex problems with high-dimensional data and non-linear decision boundaries. It maps input data into a high-dimensional space and attempts to find an optimal hyperplane that separates different classes, such as normal and attack instances. SVMs are effective in handling complex and non-linear data, making them suitable for detecting sophisticated cyber-attacks. They are based on the concept of finding an optimal hyperplane that maximally separates the data into different classes or predicts continuous values (A. Halimaa, K. Sundarakantham 2019).

Considering a data set of n points as $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$ where y_1 represents the class to which x_i belongs by taking on a value of -1 or 1. Each x_i is a p-dimensional real vector. The goal is to determine the *"maximum-margin hyperplane"* that divides the group of points for which $y_i = -1$ and the group of points for which $y_i = 1$. This is to ensure the distance between the hyperplane and the nearest x_i point from both groups is maximized. In other words, we want to find a hyperplane that acts as a decision boundary between the two classes, ensuring the largest possible gap or margin between the points of each class. This margin should be such that the hyperplane is equidistant from the nearest points in both classes. The task at hand involves a binary classification problem where the goal is to find an optimal hyperplane that achieves the best separation between the two classes while maximizing the distance to the closest data points. Any hyperplane can be expressed as the satisfied set of x as

$$W^T \times x - b = 0 \tag{4}$$

where W is the not necessarily normalization vector. The decision boundaries (positive and negative) parallel to the main hyperplanes are expressed respectively as follows:

$$w_0 + \mathbf{w}^T \times \mathbf{x}_{+ve} = 1 \tag{5}$$

$$w_0 + \mathbf{w}^T \times \mathbf{x}_{-ve} = -1 \tag{6}$$

subtracting equations (4) and (5) from each other yields

$$\mathbf{w}^T \times (\mathbf{x}_{+ve} - \mathbf{x}_{-ve}) = -1 \tag{7}$$

normalizing equation (6) by the length of the vector w defined as

$$||w|| = \sqrt{\sum_{j=1}^{m} w_j^2} \tag{8}$$

Thus, arriving at

$$\frac{\mathbf{w}^T (\mathbf{x}_{+ve} - \mathbf{x}_{-ve})}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||} \tag{9}$$

The margin to get maximized, can then be deduced from the left side of the preceding equation as the distance between the positive and negative hyperplane. A constraint must be added on either side to keep the margin at its maximum to prevent the data points from falling within the margin. This can be written as

$$w_0 + \mathbf{w}^T \times \mathbf{x}_i \geq 1, if\ y_i = 1 \tag{10}$$

$$w_0 + \mathbf{w}^T \times \mathbf{x}_i \geq 1, if\ y_i = -1 \tag{11}$$

Equations (10) and (11) aim to distinguish between negative and positive samples by utilizing hyperplanes. For the negative samples, the equation ensures that all of them fall on one side of the negative hyperplane. This means that when the negative samples are projected onto the hyperplane, they should be classified as negative by the model. Similarly, for the positive samples, the equation guarantees that they should fall behind the positive hyperplane. When the positive samples are projected

onto the positive hyperplane, the model should classify them as positive.
By formulating these conditions mathematically, the equations establish the necessary criteria for separating of negative and positive samples in the feature space. The model can then use these hyperplanes to make accurate predictions and classify future instances as either negative or positive based on their position relative to the hyperplanes. This can be written as

$$y_i(w_0 + \mathbf{w}^T \times \mathbf{x}_i) \geq 1, \textit{ for all } 1 \leq i \leq n \tag{12}$$

To get the optimization problem,

$$\begin{aligned} &\underset{\mathbf{w}, w_0}{\text{minimize}} \quad ||w||_2^2 \\ &\text{subject to} \quad y_i(x_i + \mathbf{w}^T x_j) \geq 1 \quad \forall_i \in 1, ..., n \end{aligned} \tag{13}$$

The hinge loss function is commonly used in SVM to handle cases where the data is not linearly separable. The hinge loss function introduces a notion of a *soft margin* that allows for some misclassification while still aiming to maximize the margin between the decision boundary and the data points. The hinge loss function is defined as follows:

$$max(0, 1 - y_i(w_0 + \mathbf{w}^T \times \mathbf{x}_i)) \tag{14}$$

It measures the degree of misclassification or violation of the margin constraint. If a data point is correctly classified and falls on the correct side of the margin, the hinge loss function will yield a value of zero, indicating that there is no penalty for that particular sample. The optimization's objective is to reduce.

$$\lambda ||w||^2 + \left[\frac{1}{n} \sum_{i=1}^{n} max\left(0, 1 - y_i(w_0 + \mathbf{w}^T \times \mathbf{x}_i)\right) \right] \tag{15}$$

where $\lambda > 0$ determines the trade-off between increasing the margin size and ensuring that the $\mathbf{x}_i$ falls on the correct side of the margin. This optimization problem can be reduced to the following by dissecting the hinge loss:

$$\begin{aligned} \underset{\mathbf{w}, w_0}{\text{minimize}} \quad & ||w||_2^2 + C\sum_{i=1}^{n} \zeta \\ \text{subject to} \quad & y_i(x_i + \mathbf{w}^T x_i) \geq 1 \quad \forall_i \in 1, \ldots, n \end{aligned} \tag{16}$$

We may then regulate the penalty for misclassification using the variable C. Large values of C result in significant error penalties, whereas lesser values of C result in less stringent penalties for misclassification errors.

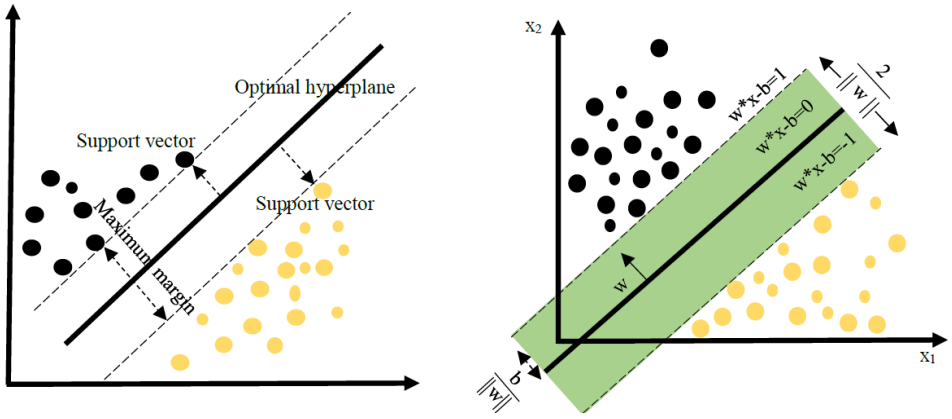


Figure 9: Support vector machine.

Extensive overviews of SVMs are:

1. Basic concept: The fundamental idea behind SVM is to find a hyperplane that best separates the data into distinct classes. In a binary classification scenario, the hyperplane is a decision boundary that maximizes the margin, which is the distance between the hyperplane and the closest data points from each class, known as support vectors. SVM aims to find the hyperplane that achieves the maximum margin while minimizing classification errors.

2. Linear SVM: Linear SVM deals with linearly separable data, where a straight line or hyperplane can separate the data into different classes. The goal is to find the hyperplane that maximizes the margin. This optimization problem is formulated as a quadratic programming problem, where the objective is to minimize the norm of the weight vector while satisfying certain constraints. The support vectors, which lie closest to the decision boundary, are critical for defining the hyperplane.

3. Non-linear SVM: In many real-world scenarios, data is not linearly separable, and a linear hyperplane cannot effectively classify the data. Non-linear

SVM addresses this by transforming the original feature space into a higher-dimensional space using the kernel trick. The kernel function computes the inner products between pairs of transformed data points without explicitly computing the transformation itself. This allows SVM to implicitly operate in a higher-dimensional space, where the data may become linearly separable. Commonly used kernel functions include linear, polynomial, Gaussian radial basis function, and sigmoid.

4. Margin and soft margin: The margin in SVM refers to the region between the decision boundary and the support vectors. SVM aims to find the hyperplane with the maximum margin, which is believed to generalize better on unseen data. However, in practice, data may not be perfectly separable, or there may be outliers. A soft margin approach is introduced to handle such cases, allowing some misclassifications and errors. The objective becomes a trade-off between maximizing the margin and minimizing the classification errors or violations of the margin.

5. *C*-parameter: The *C*-parameter in SVM controls the regularization or the balance between achieving a larger margin and allowing misclassifications. A smaller *C*-parameter results in a wider margin and tolerates more misclassifications, leading to a more robust model but potentially with lower accuracy. A larger *C*-parameter emphasizes classification accuracy and may lead to a narrower margin, potentially resulting in overfitting the training data.

6. Multi-class classification: SVM is naturally a binary classifier but can be extended to handle multi-class classification problems. One approach is to use the one-vs-rest (OvR) strategy, where multiple binary SVM classifiers are trained, each distinguishing one class from the rest. Another approach is the one-vs-one (OvO) strategy, where separate binary classifiers are trained for each pair of classes. The final prediction is made by majority voting or by considering pairwise classification results.

7. SVM regression: In addition to classification, SVM can be used for regression tasks, where the goal is to predict continuous values. The objective is to find a hyperplane that maximizes the margin while allowing a specified margin of tolerance for deviations or errors. The prediction is based on the distance of the test instance from the hyperplane.

SVM offers several advantages, including:

- Effective in high-dimensional spaces: SVM can handle datasets with a large number of features and is less susceptible to the *"curse of dimensionality"* compared to other algorithms. It is able to find complex decision boundaries and capture intricate relationships between features.

- Robust to outliers: SVM is less affected by outliers since it focuses on the support vectors closest to the decision boundary. Outliers have a minimal impact on the final hyperplane.

- Versatility through kernel functions: Using kernel functions allows SVM to handle non-linearly separable data by implicitly mapping it to a higher-dimensional space. This flexibility enables SVM to capture complex patterns and relationships.

- Global solution: SVM aims to find the optimal hyperplane that maximizes the margin. The solution is determined by the support vectors, which are a subset of the training data. Thus, the solution is not dependent on the entire dataset and is more likely to generalize well.

However, the SVM has some limitations, which include:

- Computational complexity: SVM can become computationally expensive, especially when dealing with large datasets or high-dimensional feature spaces. SVM's training time complexity is generally cubic concerning the number of training instances. However, various optimization techniques, such as efficient solvers and kernel approximations, can mitigate this issue.

- Sensitivity to parameter tuning: SVM performance heavily relies on appropriate parameter selection, including the choice of the kernel function and regularization parameter C. These parameters are often selected through cross-validation or grid search, which can be time-consuming and require expertise.

- Interpretability: The resulting model may lack interpretability, while SVM provides accurate predictions. The decision boundary is represented by a complex hyperplane in a transformed feature space, making it challenging to directly interpret the relationship between the original features and the predictions.

- Memory requirements: SVM models require storing the support vectors, which can be memory-intensive, especially when dealing with large datasets. Additionally, in the case of non-linear kernels, the kernel matrix may need to be stored, resulting in increased memory usage.

- Regularization control: The C-parameter in SVM allows users to control the balance between the margin size and the number of misclassification, providing flexibility in managing the trade-off between model complexity and training accuracy.

In conclusion, SVMs are versatile and powerful machine learning algorithms that effectively handle classification and regression tasks. They are particularly beneficial in scenarios with high-dimensional data, non-linear relationships, and the need

for robustness to outliers. However, the computational complexity and parameter tuning challenges should be considered, and interpretability may be limited in some cases. SVM remains a valuable tool in the machine learning toolkit, offering accurate predictions and the ability to capture complex patterns in the data.

3.2.3 Random forests

Random forests are like a bustling forest of decision trees, working together to create a robust and accurate prediction system. Imagine stepping into this vibrant ecosystem, where each tree has its unique character and role. As you enter the forest, you notice the diversity and randomness surrounding you. This is the key strength of random forests. Each decision tree is constructed using a random subset of the training data, creating an element of unpredictability (Mellor et al., 2015). It is as if each tree has its own story, having been trained on a different data slice. Moving deeper into the forest, you witness the individual trees at work. They independently grow and branch out, forming their own rules and decisions. These decision trees, with their branches representing different features and attribute values, capture the essence of the data they were trained on (Kulkarni & Sinha, 2012).

It is fascinating to see how each tree learns from its unique perspective, forming its understanding of the patterns and relationships within the data. But random forests are not just a collection of independent trees. They are a tightly-knit community, collaborating to make accurate predictions. The magic happens when the forest comes together, combining the wisdom of all the trees. It is like a democratic voting system, where each decision tree gets a say in the final prediction. In classification tasks, the class receiving the most votes becomes the predicted class. This voting mechanism ensures a robust and reliable prediction, as it considers the collective intelligence of the entire forest (Long et al., 2019).

Random forests are incredibly resilient to outliers and noisy data points. Outliers may influence a single decision tree, but their impact is diluted when considering the majority vote of multiple trees. The forest as a whole is less swayed by these outliers, providing a more balanced and accurate prediction. One striking feature of random forests is their ability to handle high-dimensional data (Belgiu & Dragu, 2016). It is as if these trees have developed a knack for navigating through the complexity of the forest. They effortlessly evaluate different features and select the most informative ones at each split. This adaptability allows them to capture intricate relationships, making them highly effective in scenarios where the data exhibits non-linear patterns.

Moreover, random forests are renowned for their versatility. They can seamlessly transition between classification and regression tasks. In regression, instead of predicting classes, the forest predicts numerical values based on the collective knowledge of the trees. It is like the forest whispering its combined wisdom to estimate

continuous outcomes. Each tree is crucial in this lively and dynamic forest, but the random forests' collective power shines through. The forest thrives on the principle that the whole is greater than the sum of its parts. It brings together diverse perspectives, harnesses the strength of randomness, and leverages the voting mechanism to create a robust and accurate prediction system. In conclusion, random forests are an ensemble learning technique that combines multiple decision trees to improve accuracy and robustness in IDS. Random forests create a set of decision trees using different training data and feature subsets. The final classification is based on the individual decision trees' majority vote or average prediction. Random forests effectively handle in handling high-dimensional data and can detect various types of cyber-attacks.

3.2.4 Deep learning

Deep learning is a subfield of machine learning that focuses on training artificial neural networks (ANN) with multiple layers, known as deep neural networks (DNN), to learn and extract hierarchical representations from data (Taji et al., 2018). It has revolutionized various domains, including computer vision, natural language processing, speech recognition, and many others, by achieving state-of-the-art performance on complex tasks (Jamil et al., 2021).

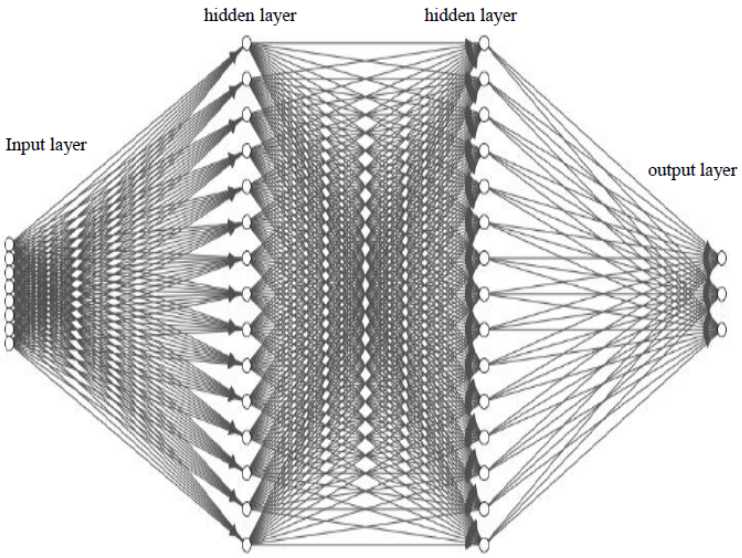


Figure 10: Description of an artificial neural network.

Deep learning heavily relies on ANNs, which are inspired by the structure and functioning of biological brains. ANNs consist of interconnected nodes called neurons, organized in layers. Each neuron receives input, performs a computation, and generates an output. Neural networks can have multiple layers, including an input layer,

one or more hidden layers, and an output layer (Buczak & Guven, 2016; Niculescu, 2003). Deep learning extends traditional neural networks by adding more hidden layers, resulting in DNN. These deep architectures allow more complex and abstract representations to be learned from the data (Sarker, 2022b).

Deep learning excels at representation learning, which involves automatically learning hierarchical representations of data. In DNNs, each layer learns progressively more abstract features by building upon the representations learned in the previous layers. The initial layers capture low-level features, such as edges or textures, while subsequent layers combine these features to represent higher-level concepts. With DNNs, the learning process becomes more efficient at capturing intricate patterns and relationships, leading to improved performance on challenging tasks. As we have inputs flowing through weighting and processing to the output layer by layer, we have a feedforward neural network that can be mathematically represented as given in (Deb et al., 2018)

$$y = f\left(\sum_{i=1}^{N} x_i w_1\right) \tag{17}$$

This hierarchical representation learning is a crucial factor behind the success of deep learning in capturing complex patterns. They are trained using a technique called backpropagation. During training, the network receives input data, makes predictions, and compares them to the true labels. The error or loss between the predicted and true labels is then backpropagated through the network, adjusting the weights and biases of each neuron to minimize the error. This is illustrated in Figure 9 (M. Elmusrati, 2022).

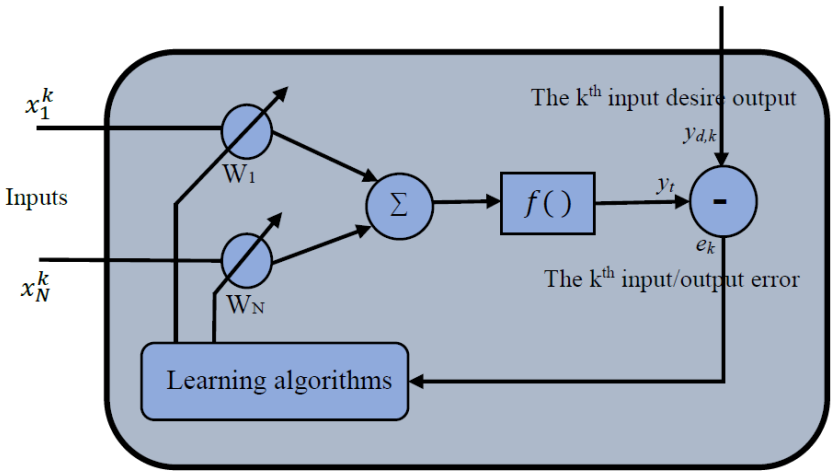


Figure 11: Illustration of single-layer neural network

Considering that $x_1^k = 1$, the w_1 is the bias (offset) of the neuron.

$$y_k = f\left(x_1^k w_1^k + x_2^k w_2^k + ... + x_N^k w_N^k\right) \tag{18}$$

The error between the desired output and the obtained output is computed as

$$e_k = y_{d,k} - y_t \tag{19}$$

The learning principle is based on adjusting the weight to reduce the cost function. The cost function is an average error-related norm. To reduce the average error between the desired and actual outputs while taking into account all learning samples. This can be written as

$$J(w^k) + E = \left[\left|e_k\right|^2\right] = E\left[\left|(y_{d,k} - y_t)\right|^2\right] \tag{20}$$

Substituting equation (18) into equation (20) yields (M. Elmusrati, 2022),

$$J(w^k) + E = E\left[\left|y_{d,k} - f\Big(\sum_{i=1}^{N} x_i^k w_i^k\Big)\right|^2\right] \tag{21}$$

This iterative optimization process, often coupled with gradient descent algorithms, fine-tunes the network's parameters to improve its predictive capabilities. Activation functions are essential in neural networks as they introduce non-linearity into the computations of each neuron. Non-linear activation functions, such as the rectified linear unit (ReLU), sigmoid, or hyperbolic tangent, enable the network to model complex relationships and capture non-linear patterns in the data. The choice of activation function influences the network's ability to learn and generalize from the data.

A ReLU is a commonly used activation function in deep learning and ANNs. It is a simple but powerful mathematical function that introduces non-linearity to the network, enabling it to learn and model complex relationships in the data. ReLu is mathematically written as (Javid et al., 2022; Qiumei et al., 2019)

$$f(x) = \begin{cases} 1 & \text{if} \quad x > 0 \\ 0 & \text{if} \quad x < 0 \end{cases} \tag{22}$$

where x represents a neuron's input.

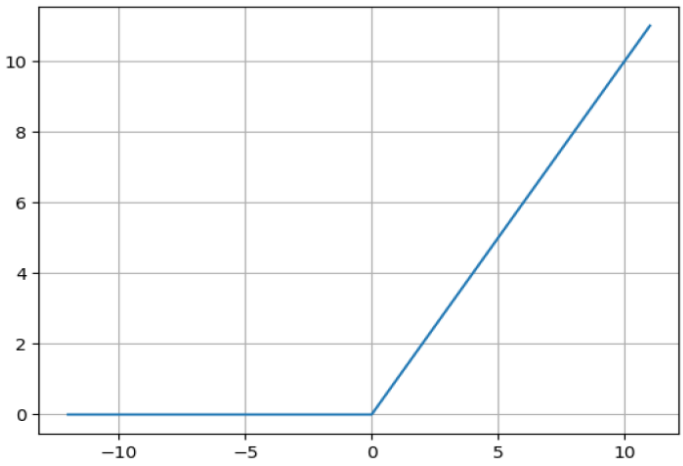


Figure 12: Description of a ReLU activation function.

The sigmoid activation function is a well-known nonlinear function in ANN. Because of its shape, which resembles an *S* curve, it is also known as the logistic function.

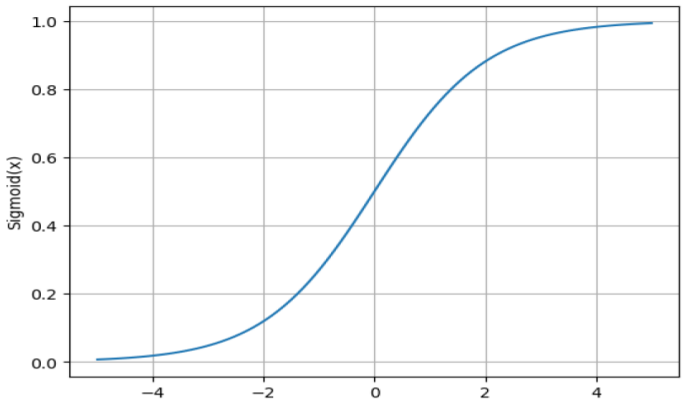


Figure 13: Description of a sigmoid activation function.

The sigmoid function is excellent for binary classification applications because it transfers every real-valued number to a value between 0 and 1. The sigmoid function is mathematically expressed as (Javid et al., 2022; Qiumei et al., 2019):

$$f(z)\frac{1}{(1+e^{-z})} \tag{23}$$

It takes an input value and transforms it using the exponential function, subsequently scaling the result to a range between 0 and 1. One of the main issues is that the sigmoid function saturates at extreme values, where the output becomes close to 0 or 1.

This saturation can cause gradients to vanish, making it difficult for the network to learn and converge. Although it has some limitations, such as saturation and vanishing gradients, it still finds its use in specific scenarios where the output needs to be bounded between 0 and 1.
The mathematical expression of the hyperbolic tangent function is given by (Qiumei et al., 2019)

$$f(x)\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{24}$$

In this formula, e represents Euler's number (2.71828), and x is the input to the function. The hyperbolic tangent function takes the difference of two exponential terms and divides them by their sum. The hyperbolic tangent function shares some similarities with the sigmoid function,

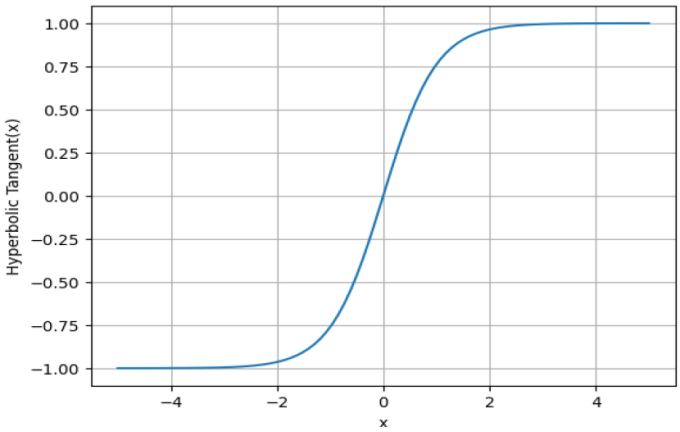


Figure 14: Description of a hyperbolic tangent activation function.

but it has a steeper slope around the origin, which means it can produce stronger and more decisive activations. Like the sigmoid function, the hyperbolic tangent function also exhibits saturation at extreme values, where the output approaches -1 or 1, leading to vanishing gradients. The advantages of using the hyperbolic tangent function include its non-linearity and its ability to capture complex relationships in the data. It is commonly used as an activation function in RNNs and certain feed-forward neural networks. The zero-centered property of hyperbolic tangent can be

beneficial in certain cases where the data distribution is symmetric around zero. However, similar to the sigmoid function, the hyperbolic tangent function may suffer from the vanishing gradient problem, especially in DNNs. When gradients become extremely small, the network's ability to learn and make meaningful updates to its weights diminishes.

Convolutional neural networks (CNNs): CNNs are a specialized type of DNN widely used in computer vision tasks. They leverage the concept of convolution, where filters are applied to local regions of the input image, allowing the network to learn spatial hierarchies of features. CNNs have proven to be highly effective in tasks such as image classification, object detection, and image segmentation. CNNs consist of convolutional layers, pooling layers, and fully connected layers.

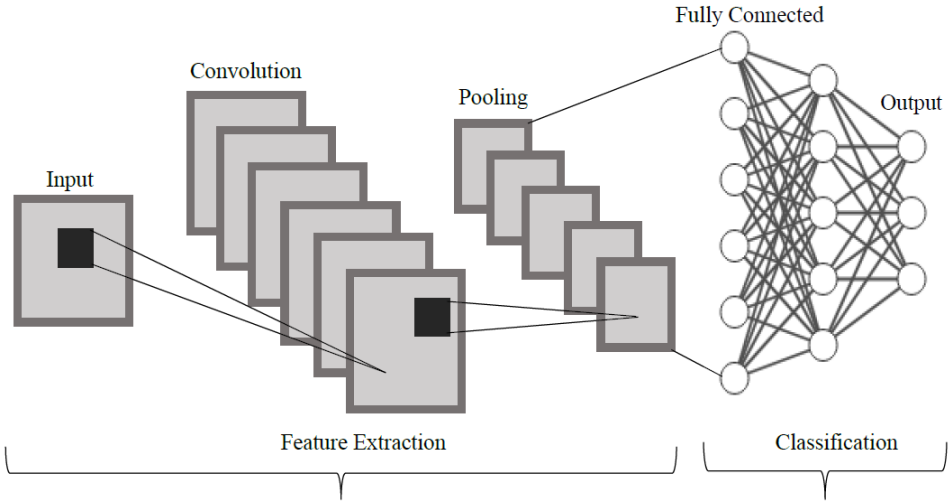


Figure 15: Convolutional neural network.

Convolutional layers learn local patterns and features from input data by applying filters or kernels. These filters capture spatially local patterns like edges and textures. Pooling layers reduce spatial dimensions while preserving important information. This pooling can be done in a variety of ways, such as by taking the mean, maximum, or a learned linear combination of the neurons in the block. Always taking the maximum of the block they are pooling. They down sample feature maps using operations like max pooling. Fully connected layers map high-level features to output classes, enabling predictions based on learned features. CNNs are trained using backpropagation, adjusting parameters to minimize the error between predicted and ground truth labels. They learn hierarchical representations, with lower layers capturing low-level features and higher layers learning complex and abstract features.

CNNs benefit from parameter sharing, where filters are applied to different image regions, reducing parameters and improving computational efficiency. Pretrained CNN models like ResNet, and Inception, trained on large datasets like ImageNet, demonstrate excellent transfer learning capabilities. These models can be fine-tuned on specific datasets with limited labeled data. CNNs have transformed computer vision because they can learn hierarchical features and achieve high accuracy. They are integral to state-of-the-art models and crucial in advancing computer vision applications.

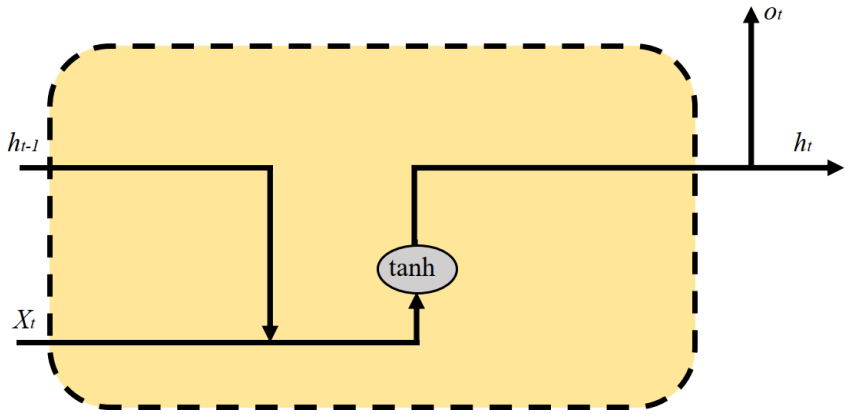


Figure 16: Recurrent neural network.

RNNs: RNNs are another important class of DNNs designed to process sequential data, such as time series or natural language. RNNs have connections that allow information to flow in loops, enabling them to capture temporal dependencies and contextual information. RNNs are a type of ANN designed to process sequential and temporal data. Unlike feedforward neural networks, which process inputs independently, RNNs have connections that form directed cycles, allowing them to maintain and propagate information across different time steps (Jozwiak et al., 2020).

The recurrent structure enables RNNs to capture dependencies and patterns in sequences, making them suitable for natural language processing, speech recognition, time series analysis, and more. The fundamental building block of an RNN is the recurrent neuron, which maintains an internal state or memory that evolves, as illustrated in Figure 16. This memory allows the network to retain information about past inputs and use it to influence future predictions. At each time step, the recurrent neuron combines an input with the previous state to produce an output and update its internal memory. This process is repeated for every time step, creating a recurrent feedback loop. Mathematically, an RNN can be defined as

$$h_t = \sigma_h\Big(W_h x_t + U_h h_{t-1} + b_h\Big) \tag{25}$$

where h_t represents the hidden state or memory at time step t, x_t is the input at time step t, σ is an activation function, W is the input weight matrix, U is the recurrent weight matrix, and b_h is the bias term. The input weight matrix W determines how the current input x_t is combined with the previously hidden state h_{t-1}, while the recurrent weight matrix U controls the influence of the previous state on the current state.

Various activation functions can be used in RNNs, such as the tanh or ReLU. These functions introduce non-linearities to the network, enabling it to learn complex relationships and capture non-linear dependencies in the data. One of the challenges with traditional RNNs is the vanishing gradient problem, which occurs when the gradients diminish exponentially as they propagate backward in time during the training process. This issue limits the network's ability to capture long-term dependencies and can hinder learning. To mitigate this problem, several variants of RNNs have been proposed, such as the long short-term memory (LSTM) (Dong et al., 2018) and gated recurrent unit (GRU) architectures. These variants incorporate gating mechanisms that allow the network to selectively update and forget information, improving the model's ability to handle long-term dependencies. LSTMs, for example, have memory cells that can retain information over long periods. They use gates, such as the input gate, forget gate, and output gate, to control the flow of information into and out of the memory cells. This gating mechanism helps address the vanishing gradient problem and allows LSTMs to capture and remember important information across multiple time steps.

Training RNNs typically involves using techniques like backpropagation through time (BPTT) or variants of it, which adapt the network's parameters based on the error between predicted and target outputs. Optimization aims to minimize the loss function by adjusting the weights and biases using optimization algorithms such as stochastic gradient descent (SGD) or its variants. RNNs can be used in various applications. They can model and generate text in natural language processing, perform sentiment analysis, and machine translation. In speech recognition, RNNs can convert spoken language into written text. They are also effective in time series analysis, forecasting, and anomaly detection. In recent years, RNNs have been further combined with other neural network architectures to create more powerful models. For example, the combination of CNNs and RNNs, known as convolutional recurrent neural networks (CRNNs), can process sequential data with both local and temporal dependencies, as in image captioning tasks.

In conclusion, the LSTM is a variant of RNNs designed to address the limitations of traditional RNNs, particularly in capturing and retaining long-term dependencies in sequential data. LSTMs have become one of the most widely used architectures for tasks involving sequential data, such as natural language processing, speech recognition, and time series analysis. The key innovation of LSTM is the introduction of memory cells, which allow the network to selectively retain and forget information over extended periods. This memory mechanism enables LSTMs to effectively capture long-range dependencies and overcome the vanishing gradient problem typically associated with traditional RNNs. At the core of an LSTM cell are three main components: the forget gate, the input gate, and the output gate. These gates regulate the flow of information into and out of the memory cell, allowing the LSTM to control the information it retains and utilizes.

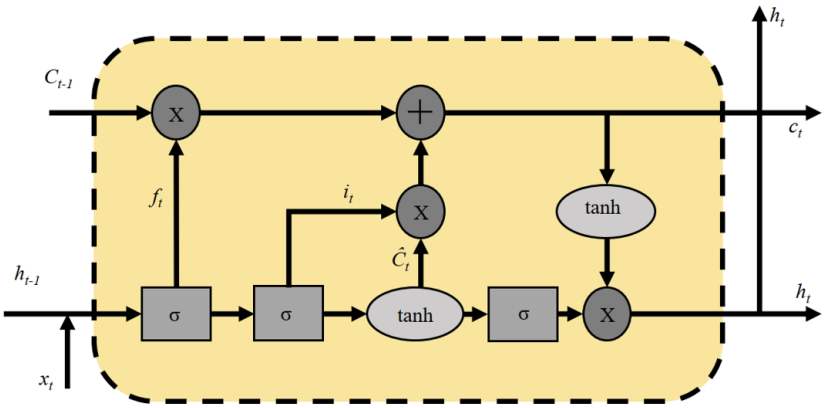


Figure 17: Long short-term memory.

Forget gate: The forget gate decides what information in the memory cell should be discarded. Similar to the input gate, it takes the current input and the previous hidden state as input and applies a sigmoid activation function. The output of the forget gate, ranging from 0 to 1, determines the amount of information to be discarded from the memory cell. A value of 0 signifies complete forgetting, while a value of 1 means the information is retained entirely.

$$f_t = \sigma\left(W_f[h_{t-1}, X_t] + b_f\right) \tag{26}$$

Input gate: The input gate determines how much new information should be stored in the memory cell. It takes the current input and the previous hidden state as input and applies a sigmoid activation function to generate a value between 0 and 1

(Rajput et al., 2021). This value represents the amount of information stored in the memory cell, with 0 indicating no new information and 1 indicating full information retention.

$$i_i = \sigma\Big(W_i[h_{t-1}, X_t] + b_i\Big) \tag{27}$$

The new memory network is a neural network trained to generate a "new memory update vector" by integrating the prior hidden state with the current input data. This vector contains information from the incoming data and considers the context provided by the preceding concealed state. The new memory update vector defines how much each long-term memory component (cell state) should be updated depending on the most recent data.

$$\hat{C}_t = \tanh\Big(W_c[h_{t-1}, X_t] + b_c\Big) \tag{28}$$

Output gate: The output gate determines how much of the memory cell's content should be exposed as the output of the LSTM cell. It takes the current input and the previous hidden state as input and applies a sigmoid activation function.

$$O_t = \sigma\Big(W_o[h_{t-1}, X_t] + b_o\Big) \tag{29}$$

The updated cell state represents the updated long-term memory of the network. The internal state is updated with this rule.

$$C_i = i_t \times \hat{C} + f_t.C_t - 1 \tag{30}$$

A tanh activation function is also applied to the current input and the previous hidden state. The output gate then multiplies the tanh output with the sigmoid output, producing the final output of the LSTM cell.

$$h_t = O_t \times \tanh(C_t) \tag{31}$$

These three gates work together to teach LSTMs to retain and use information over long sequences selectively. The LSTM may learn which information is relevant to retain, forget, or output by adjusting the weights and biases of these gates during training. Deep LSTM networks can be formed by stacking LSTMs, allowing for modeling progressively more intricate dependencies in sequential data. The output of one LSTM layer is used as the input for the next layer, allowing the network to record hierarchical data representations. Training LSTMs involves techniques such as BPTT, which propagates gradients through the entire sequence, updating the LSTM's parameters to minimize the error between predicted and target outputs. Optimization algorithms like SGD or its variants are commonly used for this purpose. LSTM is a form of RNN designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. Memory cells and gating mechanisms are used by LSTMs to selectively keep, forget, and output information. They have been widely utilized in a variety of disciplines, including natural language processing, speech recognition, and time series analysis, and have significantly increased neural networks capabilities in dealing with sequential data.

GRU is another variant of RNNs that, like LSTM, addresses the challenges of capturing long-term dependencies in sequential data. GRU was introduced as a simplified alternative to LSTM, offering comparable performance and a simpler architecture with fewer parameters. GRU retains the concept of memory cells and gating mechanisms but combines the input and forget gates of LSTM into a single update gate. This simplification allows for more efficient computation and more straightforward implementation. The main components of a GRU cell are the update gate and the reset gate. These gates control the flow of information within the cell, enabling it to selectively update or reset the internal state.

- Update gate: The update gate in a GRU cell determines how much of the past internal state should be combined with the current input. It inputs the previous hidden state and the current input and applies a sigmoid activation function. The output of the update gate, ranging from 0 to 1, represents the portion of the previous hidden state to be retained. A value of 1 means fully retaining the previous state, while a value of 0 means discarding it entirely.

- Reset gate: The reset gate regulates how much of the previous hidden state should be forgotten or reset. It inputs the previous hidden state and the current input and applies a sigmoid activation function. The output of the reset gate determines the extent to which the previous hidden state should be reset. A value of 1 signifies no resetting, while a value of 0 indicates a complete reset.

The combination of the update and reset gates allows the GRU cell to update its internal state adaptively, selectively remembering or forgetting information based on the current input and previous hidden state. It also introduces a new candidate hidden state that is computed based on the reset gate and the current input. This

candidate hidden state contains new information that can be potentially added to the updated hidden state.
Mathematically, the computations in a GRU cell can be summarized as follows (Diaba et al., 2022):

$$\Gamma_U = \sigma\left(\omega_U\left[\tilde{\xi}^{(t-1)}, x^t\right] + b_U\right) \tag{32}$$

$$\Gamma_R = \sigma\left(\omega_U\left[\tilde{\xi}^{(t-1)}, x^t\right] + b_R\right) \tag{33}$$

Representing the update gate and the reset gate are the Γ_U and Γ_R respectively. The range of the GRU gate $\in \{0,1\}$. Where ω_U stands for the weight function of the update gate and ω_R stands for the weight function of the reset gate. The bias vectors for the update and reset gates are denoted by b_U and b_R respectively. $\tilde{\xi}^{(t-1)}$ is the current unit input which is obtained from the previous unit output and x^t is the inputs of training data. Thus, the recurrent unit's candidate activation function can be written as

$$\tilde{\xi}^{(t)} = tanh\left(\omega_V\left[\Gamma_R \times \tilde{\xi}^{(t-1)}, x^t\right] + b_V\right) \tag{34}$$

where $\tilde{\xi}^{(t)}$ is the candidate activation function,ω_V is the weight functions of the activation function, x^t is the inputs of training data, and b_V is the bias vector. The output of a single GRU unit is given correspondingly as

$$\xi^{(t)} = \left((1-\Gamma_R) \times \tilde{\xi}^{(t-1)} + \left(\Gamma_R * x^t\right)\right. \tag{35}$$

GRU networks can be stacked to form deep GRU architectures, allowing for the modeling of complex sequential dependencies. The output of one GRU layer serves as the input to the next layer, facilitating hierarchical representations. Training GRU networks typically involves techniques such as BPTT, where gradients are propagated through the entire sequence to update the network's parameters.
Optimization algorithms like SGD or its variants, such as Adam or RMSprop, are commonly employed in training GRU networks. GRUs have shown strong performance in various sequential data applications, such as natural language processing, speech recognition, and time series analysis. They offer a balance between modeling capability and computational efficiency, making them a popular choice when capturing long-term dependencies is crucial.

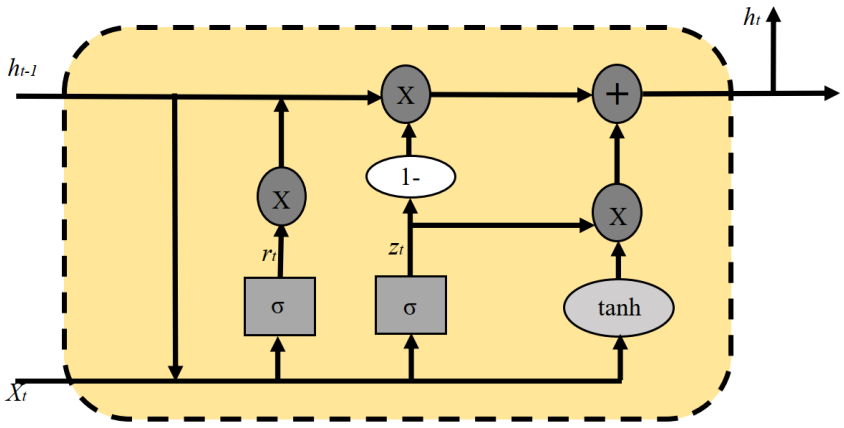


Figure 18: Gated recurrent unit.

In summary, GRU is a variant of RNNs that addresses the challenge of capturing long-term dependencies in sequential data. GRU simplifies the architecture by combining the input and forget gates of LSTM into a single update gate and introducing a reset gate (de Souza et al., 2022). The update gate controls how much of the previous hidden state should be retained, while the reset gate determines the extent to which the previous hidden state should be reset. This allows the GRU cell to adaptively update its internal state, selectively remembering or forgetting information based on the current input and previous hidden state. GRUs have demonstrated strong performance in various applications, including natural language processing, speech recognition, and time series analysis. They offer a balance between computational efficiency and modeling capability, making them a popular choice for tasks involving sequential data.

3.2.5 Linear regression

Linear regression is a fundamental statistical technique that aims to model the relationship between a dependent variable and one or more independent variables. It belongs to the broader field of regression analysis and is widely used in various domains, including economics, finance, social sciences, and machine learning. At its core, linear regression assumes a linear relationship between the dependent and independent variables. This implies that a straight line in a scatter plot can represent the relationship.

The goal of linear regression is to estimate the parameters of this line to best fit the observed data. The dependent variable, also known as the target variable or response variable, is the variable being predicted or explained.

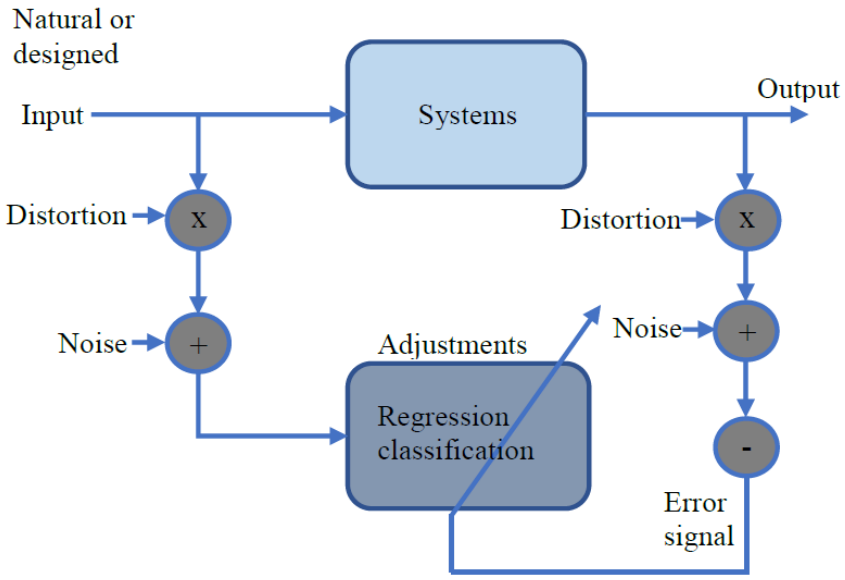


Figure 19: The concept of regression.

The independent variables, also called predictors or explanatory variables, are the variables used to predict or explain the dependent variable.
Given a dataset $\{y_i, x_i, ..., x_{i=1}^n\}$ of n statistical units, a linear regression model presupposes that the relationship between the dependent variable y and the vector of regressors x is linear. The relationship can be modeled as

$$y = \alpha x + \beta \tag{36}$$

Where x is the input, y is the output, α represents the slope coefficient (the coefficient associated with the independent variable x), and β represents the y-intercept (the constant term). There are two primary types of linear regression: simple linear regression and multiple linear regression. Simple linear regression involves a single independent variable, whereas multiple linear regression incorporates two or more independent variables. The distinction lies in the number of predictors used to explain the dependent variable. The multiple linear regression formula extends the concept of one independent to include multiple independent variables as

$$y = \beta + \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \tag{37}$$

The linear regression model seeks to find the best-fit line that minimizes the dif-

ference between the predicted values and the actual values of the dependent variable. This is termed an error with an error vector $[e_1, e_2, ...e_n]$. This difference is often measured using a metric called the residual, which represents the vertical distance between the observed data points and the line. To estimate the parameters of the line, the model uses a method called the least squares estimation (LSE). This method calculates the sum of the squared residuals and adjusts the line parameters to minimize this sum. The model aims to find the line that best fits the data by minimizing the sum of squared residuals.

$$\xi = \frac{1}{n}\sum_{i=1}^{n} e_1^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y_1}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \alpha x_i - \beta\right)^2 \tag{38}$$

Since the goal is to minimize the mean square error values, the Karush-Kuhn-Tucker (KKT) conditions for the minimum point are applied as

$$\frac{\partial \xi}{\partial \alpha} = 0$$

and

$$\frac{\partial \xi}{\partial \beta} = 0$$

Thus, solving the first derivative yields (M. Elmusrati, 2022)

$$\frac{\partial \xi}{\partial \alpha} = -\frac{2}{n}\sum_{i=1}^{n} x_i\left(y_i - \alpha x_i - \beta\right) = 0 \Rightarrow \alpha\sum_{i=1}^{n} x_1^2 + \beta\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \tag{39}$$

$$\frac{\partial \xi}{\partial \beta} = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \alpha x_i - \beta\right) = 0 \Rightarrow \alpha\sum x_i + n\beta = \sum y_i \tag{40}$$

Equations (38) and Equation (39) in the matrix notation can be written as

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{bmatrix} \tag{41}$$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{bmatrix} \tag{42}$$

It is important to note that linear regression assumes several underlying assumptions to provide reliable results. These assumptions include linearity, independence of errors, homoscedasticity/ homogeneity of variances (constant variance of residuals), and normality of errors. Violations of these assumptions can affect the validity and accuracy of the regression model.

3.2.5.1 Logistic regression

Logistic regression is a statistical model used to predict the probability of an event occurring or not occurring in binary classification tasks. It is an extensively utilized and interpretable machine learning and statistics algorithm (Phom et al., 2010). Logistic regression is used in numerous disciplines, including machine learning, medical fields, and the social sciences. Logistic regression is an extension of linear regression that uses a logistic function, also known as the sigmoid function as given in Equation (23), to account for binary outcomes. The dependent variable or target variable in logistic regression is binary or dichotomous, meaning it can take on one of two values. Typically, these values are represented as 0 and 1, or occasionally as negative and positive classes. The independent variables, also known as features or predictors, can be categorical or continuous. The logistic regression model implies a linear relationship between independent variables and the target variable's log odds.

$$P(y=1|x)\frac{1}{(1+e^{-x})} \tag{43}$$

$$P(y=0|x)\frac{1}{(1+e^{-x})} \tag{44}$$

Where $P(y=1|x)$ represents the probability of the dependent variable y being one given the predictors x and $P(y=0|x)$ represents the probability of the dependent variable y being zero given the predictors x. Referring to Equation (23) and assuming that z is the linear combination $(\beta+\alpha_1 x_1+\alpha_2 x_2+...+\alpha_n x_n)$ of the predictors and their corresponding coefficients, substituting z into Equation (43) yields

$$P(y=1|x)\frac{1}{(1+e^{\beta+\alpha_1 x_1+\alpha_2 x_2+...+\alpha_n x_n})} \tag{45}$$

The log-odds, also known as the logit function, is the natural logarithm of the probability that the objective variable is equal to 1. The logit (log odds) equals the probability that an event will occur divided by the probability that it will not occur.

$$logit(P) = In\ \frac{P(y=1)}{1-P(y=1)} = \beta + \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \tag{46}$$

exponentiating both sides result

$$\frac{P(y=1)}{1-P(y=1)} = e^{\beta+\alpha_1 x_1+\alpha_2 x_2+...+\alpha_n x_n} \tag{47}$$

The exponential function of the linear regression expression is identical to the probability that the dependent variable equals a case, given some linear combination x of the predictors. This shows how the logit acts as a link function between the probability and the equation for the linear regression. The logit gives an appropriate criterion on which to execute linear regression, and the logit is easily transformed back into the odds due to its range between zero and infinity. Thus, we write the odds of the dependent variable matching a case as

$$odds = e^{\beta+\alpha_1 x_1+\alpha_2 x_2+...+\alpha_n x_n} \tag{48}$$

For a continuous independent variable, the odds ratio can be de determined as

$$OR = \frac{odds(x+1)}{odds(x)} \frac{\frac{P(x+1)}{1-p(x+1)}}{\frac{p(x)}{1-p(x)}} = \frac{e^{\beta+\alpha_1(x+1)}}{e^{\beta+\alpha_1 x}} \tag{49}$$

subjecting Equation (48) to the exponential rule $\frac{e^{\mu}}{e^{\eta}}$, Equation (48) can be rewritten as

$$\frac{odds(x+1)}{odds(x)} = e^{\alpha_1 x_1} \tag{50}$$

Logistic regression offers several benefits for binary classification problems. It is a straightforward and interpretable model that can shed light on the relationship be-

tween predictors and the dependent variable. The coefficients can be interpreted as the quantity of change in the target variable's log odds for a one-unit change in the corresponding predictor, with all other predictors held constant. This interpretability is particularly advantageous when communicating the results to non-technical stakeholders.

In addition, logistic regression can handle both continuous and categorical predictors. Typically, categorical predictors are embedded as dummy variables, with each category represented as a binary variable. This enables the logistic regression model to incorporate categorical data. Once the model has been trained, it can be used to predict new data by computing the predicted probabilities using the logistic function. Using a certain threshold, probabilities can be converted into class designations. Generally, a threshold of 0.5 is employed, with predicted probabilities above 0.5 being classified as positive and those below 0.5 as negative. However, optimizing the threshold value is important. It depends on the different risks of each class as well as the imbalance between available training data in each class.

Nevertheless, logistic regression has limitations as well. It presupposes a linear relationship between the predictors and the log odds, which may not be appropriate in all situations. The model may not effectively represent the complex data patterns if the relationship is nonlinear. In such instances, alternative nonlinear models, such as decision trees or neural networks, may be more appropriate. Logistic regression assumes the independence of observations, meaning that each data point is presumed independent of the others. It may not yield accurate results if there is a correlation or dependence between the observations, as in time series or clustered data. Specialized models such as generalized estimating equations (GEE) or mixed-effects models may be preferable in such situations.

Furthermore, logistic regression assumes that the predictors have a linear relationship with the log odds. If higher-order interactions or nonlinear relationships exist between the predictors and the dependent variable, logistic regression may not effectively capture them. Techniques such as polynomial terms, interaction terms, and basic functions can be used to introduce non-linearity into a model, but coefficient interpretation becomes more complicated. Noting that logistic regression is susceptible to overfitting if the model is too complex relative to the available data is essential. This is overfitting when the model learns noise or random fluctuations in the training data instead of the underlying patterns. Regularization techniques such as *L1* regularization (Lasso) and *L2* regularization (Ridge) can be used to reduce overfitting by incorporating a penalty term into the loss function, which discourages excessive coefficient values.

In practice, logistic regression is utilized in numerous fields, such as healthcare, finance, marketing, engineering, and the social sciences. It can be used to predict the likelihood of disease occurrence based on patient characteristics, determine the

probability of default in credit risk analysis, predict customer attrition in marketing, predict the likelihood of cyber-attack, and assess the influence of socio-economic factors on educational outcomes.

3.3 Machine learning applications in cybersecurity

Machine learning has become a potent tool in many fields, and cybersecurity is one of them. It has significantly advanced this discipline. Traditional rule-based approaches are frequently insufficient to identify and stop sophisticated attacks due to the complexity and number of cyber threats that are on the rise. By enabling automated analysis of massive volumes of data, the detection of abnormalities, and the discovery of patterns that may suggest a hostile activity, machine learning techniques have the potential to strengthen cybersecurity defenses. Several important uses of machine learning in cybersecurity are listed below.
IDS: IDS is crucial for identifying and responding to network attacks. Traditional rule-based IDS often struggle to keep pace with sophisticated attack techniques. Machine learning algorithms, particularly supervised learning methods such as SVM, random forests, and neural networks, have successfully detected various attacks. These algorithms can learn from labeled datasets, capturing the characteristics of both known and emerging threats. Machine learning-based IDS can identify anomalies and flag potentially malicious activities by continuously analyzing network traffic patterns, enabling prompt response and mitigation.

Malware detection: Malware continues to be a significant threat, constantly evolving to evade detection. Machine learning techniques offer effective solutions for malware detection. Feature-based approaches extract relevant attributes from files or code snippets and use them as input to machine learning algorithms. These algorithms can learn to distinguish between malicious and benign files, leveraging labeled datasets of known malware samples. Moreover, behavior-based approaches analyze the actions of programs or processes, identifying malicious behavior patterns indicative of malware. Machine learning models like hidden Markov models (HMM) or RNNs can learn from these patterns and detect new and unknown malware variants.

Anomaly detection: Anomaly detection is crucial for identifying suspicious activities or deviations from normal behavior. Machine learning provides effective tools for anomaly detection in cybersecurity. Unsupervised learning algorithms, including clustering, autoencoders, and one-class SVM, can learn the normal patterns in data and flag instances that deviate significantly from those patterns. This approach lets organizations detect novel attacks or unusual activities that traditional rule-based systems may miss (Ashok et al., 2017). Anomaly detection is applied to network traffic analysis, system logs, user behavior monitoring, and other cybersecurity-relevant data sources. Spam and phishing detection: Machine learning plays a vital role in combating the ever-growing problem of spam emails

and phishing attacks. Supervised learning algorithms, such as naive Bayes, decision trees, and logistic regression, are commonly used to classify emails as spam or legitimate. These algorithms learn from labeled datasets containing examples of spam and non-spam emails, capturing patterns and features indicative of spam. Furthermore, natural language processing techniques are employed to analyze email content and identify phishing attempts by detecting malicious URLs, suspicious attachments, or social engineering techniques. Machine learning models can adapt and evolve to tackle evolving spam and phishing tactics.

User and entity behavior analytics (UEBA): UEBA leverages machine learning to detect anomalies in user activities and identify potential insider threats or compromised accounts. Machine learning algorithms can establish baselines of normal behavior for each user by monitoring user behavior patterns. Deviations from these baselines, such as unusual access patterns, abnormal data transfers, or unauthorized privilege escalations, can trigger alerts. UEBA systems employ various machine learning techniques, including clustering, sequence mining, and anomaly detection, to identify patterns indicative of malicious or suspicious activities. Threat intelligence and predictive analytics: Machine learning enables organizations to leverage vast amounts of threat intelligence data to predict and proactively defend against cyber threats. Machine learning models can identify patterns and trends by analyzing historical data, anticipating potential vulnerabilities, and predicting future attacks. Predictive analytics can help organizations prioritize security measures, allocate resources effectively, and develop proactive defense strategies.

Network traffic analysis: Machine learning algorithms are employed for network traffic analysis, enabling organizations to detect and respond to network-based attacks and anomalies. Machine learning algorithms can extract meaningful features and patterns by analyzing network packets, flow data, or log files. These algorithms can learn to identify malicious activities, such as DoS attacks, port scanning, or command and control communications. Machine learning-based network traffic analysis enhances the ability to detect and mitigate threats in real time, providing a proactive defense against network-based attacks.

Vulnerability assessment and penetration testing: Machine learning techniques can be applied to vulnerability assessment and penetration testing processes. By analyzing historical vulnerability data and exploit information, the models can prioritize vulnerabilities based on their potential impact and likelihood of exploitation. These models can help security teams focus their efforts on critical vulnerabilities and perform targeted penetration testing to identify potential weaknesses in the system. Machine learning also aids in automating vulnerability scanning and reducing false positives, improving the efficiency and effectiveness of security assessments.

Security log analysis: Machine learning algorithms excel at analyzing large volumes of security log data generated by various systems, including firewalls, IDS,

and authentication logs. Organizations can identify log data patterns, correlations, and anomalies by applying machine learning techniques, such as clustering or classification algorithms. This analysis helps detect security incidents, identify indicators of compromise, and understand the context of security events (Radoglou-Grammatikis & Sarigiannidis, 2019). Machine learning also enables the development of intelligent security information and event management (SIEM) systems that can automate log analysis and provide real-time alerts for potential security breaches.

Fraud detection: In the finance and e-commerce sectors, machine learning is extensively used for fraud detection. The algorithms can identify fraudulent activities, such as credit card fraud, identity theft, and account takeover, by analyzing transactional data, user behavior, and historical patterns. Supervised learning models, anomaly detection techniques, and ensemble methods are commonly employed to develop fraud detection systems. These systems continuously learn from new data, adapting to evolving fraud techniques and minimizing false positives.

Threat hunting and incident response: Machine learning plays a significant role in threat hunting and incident response. By analyzing diverse data sources, such as network logs, endpoint data, or threat intelligence feeds, machine learning algorithms can identify potential threats and indicators of compromise. These algorithms can automate the correlation and analysis of multiple data points, enabling security teams to proactively hunt for threats and respond effectively to security incidents. Machine learning-powered incident response systems can accelerate incident triage, reduce response times, and provide actionable insights for remediation. It is important to note that while machine learning brings significant advancements to cybersecurity, it is not a silver bullet solution. Domain expertise, human analysis, and a layered defense approach are crucial in conjunction with machine learning techniques to ensure comprehensive and effective cybersecurity. Moreover, the ethical implications, interpretability, and potential vulnerabilities of machine learning models in the context of adversarial attacks must be carefully considered and addressed in the cybersecurity landscape.

3.4 Risk analysis of machine learning applications

Risk analysis of machine learning applications is crucial to ensuring the responsible and effective deployment of machine learning models in various domains. While machine learning brings tremendous opportunities, it also introduces potential risks and challenges that need to be carefully evaluated and managed. The following are some crucial considerations for machine learning application risk analysis.
Data quality and bias: One of the primary risks in machine learning is related to the quality and bias present in the training data. Machine learning models heavily rely on data for training, and the quality of the data used can significantly impact their performance. Inaccurate, incomplete, or biased data can lead to biased predic-

tions or erroneous outcomes. It's important to assess the quality of the training data to mitigate potential risks. Biased or incomplete datasets can lead to biased models, perpetuating unfair or discriminatory outcomes. It is essential to assess the training data's representativeness, diversity, and accuracy to ensure fair and unbiased predictions. Rigorous data preprocessing, data cleaning, and careful feature selection can help mitigate this risk.

Overfitting and generalization: Overfitting occurs when a machine learning model performs well on the training data but fails to generalize to new, unseen data. Overfit models can result in poor performance and erroneous predictions in real-world scenarios. It is critical to assess and manage the risk of overfitting by utilizing appropriate model validation techniques, such as cross-validation and regularization methods. Regularization techniques like *L1* or *L2* regularization can help control the complexity of the model and reduce overfitting.

Security and privacy: Machine learning applications often deal with sensitive data, including personal information, financial records, or proprietary business data. Security breaches or data leakage risks must be carefully evaluated and addressed. Robust security measures, data encryption, access controls, and privacy-preserving techniques like differential privacy should be implemented to protect sensitive data from unauthorized access or misuse. **Interpretability and explainability**: Machine learning models, particularly complex ones like DNNs, can be opaque and challenging to interpret. A lack of interpretability can hinder understanding model decisions, making identifying potential biases, errors, or undesirable behaviors difficult. Risk analysis should consider the need for model interpretability and explore techniques like feature importance analysis, model-agnostic methods, or rule extraction approaches to enhance the transparency and explainability of the model.

Adversarial attacks: Machine learning models are susceptible to adversarial attacks, where malicious actors intentionally manipulate or deceive the model's input to produce incorrect or malicious outputs. Adversarial attacks can have severe consequences, especially in critical domains such as healthcare, finance, or autonomous systems. Risk analysis should encompass evaluating potential vulnerabilities, understanding attack vectors, and exploring robust defense mechanisms like adversarial training, input sanitization, or anomaly detection to mitigate the risk of adversarial attacks.

Ethical and legal considerations: Machine learning applications raise ethical and legal concerns, including issues related to bias, fairness, transparency, and accountability. Evaluating the potential ethical implications of deploying machine learning models and ensuring compliance with relevant regulations, such as data protection laws (e.g., GDPR), anti-discrimination laws, or industry-specific guidelines, is crucial. Risk analysis should include assessing the machine learning application's social impact, unintended consequences, and potential legal ramifications.

System robustness and resilience: Machine learning models are typically developed and trained in controlled environments. However, real-world scenarios can introduce uncertainties, adversarial conditions, or concept drift, challenging the performance and robustness of the models. Risk analysis should consider the potential risks associated with model failures, system resilience, and the ability to handle novel or unforeseen situations. Techniques like ensemble models, anomaly detection, or continuous model monitoring can enhance system robustness and resilience.

Governance and human oversight: Machine learning applications should be governed by robust policies, guidelines, and human oversight. Risk analysis should encompass evaluating the organizational structure, accountability frameworks, and decision-making processes surrounding the use of machine learning models. Clearly defined roles and responsibilities, human-in-the-loop approaches, and mechanisms for monitoring and auditing the performance and outcomes of machine learning models are essential to ensure responsible and accountable use. **Scalability and performance**: As machine learning applications often deal with large datasets and computationally intensive algorithms, scalability and performance risks should be considered. Risk analysis should evaluate the scalability of the infrastructure and the ability to handle increased data volumes or model complexity. Additionally, the computational resources required for training and inference should be carefully assessed to ensure efficient and cost-effective operations.

Change management and model updates: Machine learning models are not static entities and may require updates or retraining over time. Risk analysis should consider the challenges associated with change management, version control, and the impact of model updates on existing systems and processes. Proper validation, testing, and deployment strategies should be in place to manage the risks associated with model updates and ensure a smooth transition. Vendor or third-party risks: In cases where machine learning applications rely on external vendors or third-party services, additional risks may arise. Assessing the reputation, reliability, and security measures of the vendors or service providers is crucial. Contracts and agreements should include provisions for data protection, intellectual property rights, service-level agreements, and mechanisms for addressing potential breaches or disputes.

Compliance and regulatory risks: Machine learning applications in regulated industries like healthcare or finance are subject to compliance requirements and industry-specific regulations. Risk analysis should consider the potential risks associated with non-compliance, such as legal penalties, reputational damage, or loss of trust. Ensuring compliance with relevant regulations, maintaining proper documentation, and conducting regular audits are vital to mitigate compliance risks.

In summary, risk analysis of machine learning applications involves a comprehen-

sive assessment of various factors, including data quality and bias, overfitting, security and privacy, interpretability, adversarial attacks, ethical and legal considerations, system robustness, governance and human oversight, scalability, change management, vendor risks, and compliance. By proactively identifying and managing these risks, organizations can foster responsible and effective deployment of machine learning models, ensuring the desired outcomes while minimizing potential pitfalls. Considering a simplified model for assessing the risk of bias in a machine learning model due to imbalanced training data.
Assumptions:

- We have a binary classification problem, where the machine learning model predicts whether an applicant will be *approved* or *rejected* based on their credit history.
- In the field of cybersecurity; a machine learning model is utilized to predict whether a class can be categorized as *attack* or *bening* specific measurements.
- In medicine; considering various measurements, should the physician determine whether the patient's symptoms indicate illness A or B?
- Based on some measurements, should the seismic system monitoring association give an alarm or not?
- The training dataset consists of historical credit data with two classes: *approved* and *rejected*

The mathematical model of the class imbalance ratio (CIR) can be defined as the ratio of the number of instances in the minority class (rejected) to the number of instances in the majority class (approved)

$$CIR = \frac{\text{Instances in the minority class}}{\text{Instances in the majority}} \tag{51}$$

To assess the risk of bias in the machine learning model due to data imbalance, we can calculate a metric called the bias risk score (BRS). The BRS is a metric that represents the potential bias introduced by the imbalanced data. It quantifies the level of risk that the model's predictions might be biased due to the class imbalance. Mathematically written as

$$BRS = \frac{CIR}{(1+CIR)} \tag{52}$$

When the CIR is close to 0, the BRS approaches 0. This means that the risk of bias due to data imbalance is relatively low. As the CIR becomes larger, the BRS also increases. This indicates that the potential risk of bias due to data imbalance is higher. The formula suggests that the BRS becomes more significant as the class imbalance becomes more extreme. It's a simple mathematical transformation that takes into account the relationship between the class imbalance ratio and the potential bias risk. The BRS ranges from 0 to 1, a BRS of 0 basically suggests that the way the classes are distributed won't likely introduce bias in the predictions, regardless of whether the classes are balanced or not. A BRS of 1 indicates a significant risk of bias due to data imbalance. In simpler terms, it means that the way the classes are unevenly distributed could strongly affect the fairness of the predictions. The higher the BRS goes toward 1, the more likely it is that the model's predictions might be unfairly influenced by the imbalanced class distribution. By assessing the calculated BRS, we can establish risk categories to evaluate the level of bias risk more effectively:

- $BRS \leq 0.2$ is categorized as low BRS, indicating that the data imbalance is minimal, and the risk of bias is low.
- $BRS \geq 0.2 \leq 0.5$ means there is some degree of data imbalance, indicating a moderate risk of bias.
- $BRS \geq 0.5$ indicating that the data imbalance is significant, suggesting a high risk of bias in the machine learning model.

This mathematical model quantitatively assesses the bias risk associated with imbalanced training data. By calculating the BRS and categorizing the risk level, informed decisions regarding data collection, preprocessing techniques, or algorithmic adjustments to mitigate the potential bias and ensure fair and accurate predictions can be made. However, when the assumption becomes

- We have an N multi-class classification problem, where the machine learning model is trained to predict the class or category of x input data based on M features. The goal is to assign each instance or sample to one and only one class out of a set of N possible classes. The probability of a feature belonging to a specific class i is denoted as (M. Elmusrati, 2022)

$$P = (C_i|x_1, x_2, ...x_M), \forall_i = 1, ..., N \tag{53}$$

The probability value assigned to a feature offers valuable insights into its correlation with a particular class i out of the N available classes. By analyzing and comparing these probabilities across different classes, we can assess the relevance and importance of the feature for each class. This analysis allows us to understand

the degree to which the feature contributes to the classification decision and helps us make informed judgments about its significance in relation to each class. Given the probability that a feature belongs to class i, by applying the Bayesian theorem and with the assumption that the feature is a discrete value, then (M. Elmusrati, 2022):

$$P = (C_i|x_1,x_2,...x_M)\frac{P(x_1,x_2,...,x_M|C_i)P(C_i)}{P(x_1,x_2,...,x_m)}, \forall_i = 1,...,N \tag{54}$$

where $P = (C_i|x_1,x_2,...x_M)$ is the posterior probability, $P(x_1,x_2,...,x_M|C_i)$ is the likelihood, $P(C_i)$ denotes class priori probability, and represents evidence. Computing Equation (54) is often difficult due to the unknown interactions between features. To tackle this issue, an assumption of attribute independence is crucial. By assuming independence among the attributes, we simplify the calculations and make the problem more tractable. This assumption allows us to treat each feature separately and focus on their individual contributions to the classification process, rather than trying to model complex interdependencies between the features. Equation (54) can thus, be manipulated as (M. Elmusrati, 2022)

$$P = (C_i|x_1,x_2,...x_M)\frac{P(x_1,x_2,...,x_M|C_i)P(C_i)}{P(x_1),P(x_2),...,P(x_m)}, \forall_i = 1,...,N \tag{55}$$

3.4.1 Algorithmic risks in machine learning

Algorithmic risks in machine learning refer to the potential challenges, limitations, and undesirable outcomes that can arise from the design, implementation, and usage of machine learning algorithms. These risks stem from various factors, including algorithmic choices, data characteristics, model assumptions, and the interaction between the algorithm and its environment. Here are some critical algorithmic risks in machine learning:

- Bias and discrimination: Machine learning algorithms can exhibit bias and discrimination if the training data reflects existing group biases or if the algorithm introduces bias during the learning process. Biased algorithms can lead to unfair or discriminatory outcomes, particularly in sensitive domains such as hiring, lending, or criminal justice. Careful attention should be given to identifying and mitigating bias, ensuring fairness and ethical decision-making.

- Overfitting and underfitting: Overfitting occurs when a machine learning model learns the training data too well, resulting in poor generalization to new, unseen data. Conversely, underfitting happens when the model fails to

capture the underlying patterns in the data. Both overfitting and underfitting can lead to suboptimal performance and inaccurate predictions. Proper model validation techniques, regularization methods, and dataset partitioning strategies are crucial to mitigate these risks.

- Lack of interpretability: Many advanced machine learning algorithms, such as DNNs, are inherently complex and lack interpretability. Interpreting how and why a model arrived at a particular decision or prediction can be challenging. A lack of interpretability can hinder transparency, trust, and identifying potential biases or errors. Techniques like model-agnostic interpretability methods, rule extraction, or explainable AI approaches can help address this risk.

- Adversarial attacks: Machine learning models can be vulnerable to adversarial attacks, where malicious actors intentionally manipulate or deceive the model's input to produce incorrect or malicious outputs. Adversarial attacks can have severe consequences, particularly in security-sensitive domains or safety-critical applications. Adversarial training, robust optimization, and input sanitization techniques are commonly employed to enhance model resilience against such attacks.

- Adversarial attacks: Machine learning models can be vulnerable to adversarial attacks, where malicious actors intentionally manipulate or deceive the model's input to produce incorrect or malicious outputs. Adversarial attacks can have severe consequences, particularly in security-sensitive domains or safety-critical applications. Adversarial training, robust optimization, and input sanitization techniques are commonly employed to enhance model resilience against such attacks.

- Model transparency and explainability: In some applications, such as healthcare or finance, it is crucial to provide explanations or justifications for the decisions made by machine learning models. Lack of transparency or explainability can hinder user acceptance, regulatory compliance, and identifying and rectifying potential errors or biases. Developing interpretable models or post-hoc explanation techniques can help mitigate this risk.

- Concept drift and model degradation: Machine learning models assume that the underlying data distribution remains stationary. However, real-world data can exhibit concept drift, where the statistical properties of the data change over time. Concept drift can lead to model degradation and deteriorating performance. Continuous monitoring, model retraining, and adaptation techniques are necessary to address this risk and maintain model effectiveness.

Addressing these algorithmic risks requires a combination of careful algorithm selection, appropriate data preprocessing, model validation techniques, robustness measures, and ongoing monitoring and maintenance. Understanding and managing these risks is essential to ensure responsible and effective deployment.

3.4.2 Algorithmic risks management

1. Risk identification: Identify and document the specific algorithmic risks relevant to the machine learning application. This involves understanding the characteristics of the algorithm, potential biases, vulnerabilities to attacks, interpretability challenges, scalability issues, and other risks associated with the chosen algorithm.

2. Risk assessment: Quantify and evaluate the severity and likelihood of each identified risk. Assess the potential impact on different stakeholders, such as end-users, decision-makers, or affected individuals. Prioritize risks based on their significance, potential consequences, and the application context.

3. Risk mitigation strategies: Develop and implement risk mitigation strategies to address the identified algorithmic risks. These strategies may include:

    - Data quality and preprocessing: Ensure data quality by performing thorough data cleaning, outlier detection, and handling missing values. Apply appropriate preprocessing techniques to address data biases, imbalances, or noise.
    - Bias and fairness mitigation: Apply fairness-aware techniques to detect and mitigate biases in the training data and algorithmic decisions. Explore approaches such as bias-correction methods, fairness-aware learning, or demographic parity to ensure equitable outcomes.
    - Security and robustness: Implement security measures to protect against adversarial attacks. Use techniques like adversarial training, robust optimization, or anomaly detection to enhance model resilience against manipulation or exploitation.
    - Explainability and interpretability: Employ techniques for model interpretability, such as feature importance analysis, rule extraction, or model-agnostic methods, to enhance transparency and explainability of algorithmic decisions.
    - Continuous monitoring and maintenance: Establish mechanisms for monitoring the model's performance, data drift, and potential risks. Regularly update and retrain the model to adapt to changing conditions and ensure continued effectiveness.

4. Validation and testing: Thoroughly validate and test the machine learning model to assess its performance, robustness, and adherence to desired specifications. Use appropriate validation techniques, such as cross-validation or holdout sets, and conduct sensitivity analyses to evaluate the model's behavior under different conditions.

5. Ethical considerations: Integrate ethical considerations into the algorithmic risk management process. Ensure compliance with applicable regulations and

guidelines related to data protection, privacy, fairness, and transparency. Establish principles for responsible AI development and use, including ongoing monitoring and assessment of the societal impact of the algorithm.

6. Documentation and accountability: Maintain comprehensive documentation of the algorithmic risk management process, including risk assessments, mitigation strategies, validation results, and ongoing monitoring activities. Assign clear roles and responsibilities to individuals or teams responsible for managing algorithmic risks. Foster a culture of accountability and transparency within the organization.

7. Stakeholder engagement and communication: Engage stakeholders, including end-users, domain experts, legal and compliance teams, and ethicists, in the algorithmic risk management process. Communicate the machine learning model's risks, mitigation strategies, and limitations effectively to ensure understanding, trust, and informed decision-making.

Algorithmic risk management is an ongoing process that requires continuous monitoring, assessment, and adaptation as new risks emerge or the operating environment evolves. Organizations can effectively manage algorithmic risks and promote responsible and trustworthy AI systems by integrating risk management practices into the machine learning development lifecycle.

4 METHODOLOGY OF MACHINE LEARNING IN CYBER SECURITY OF SMART GRIDS

4.1 Introduction

This chapter focuses on the methodology of machine learning in the cybersecurity of smart grids. Essential points covered include data collection, data cleaning and preprocessing, feature selection techniques, feature extraction, performance evaluation of machine learning models, evaluation metrics, experimental setup, model selection, model training, hyperparameter tuning, and model evaluation. The chapter aims to provide an overview of the essential steps in applying machine learning techniques to enhance the security of smart grids against cyber threats.

4.2 Data preprocessing and feature selection

4.2.1 Data collection

Comprehensive datasets from diverse sources in the field of cybersecurity were utilized for conducting experiments in smart grid infrastructure. The datasets aimed to capture various aspects of the smart grid's operations and included historical network traffic records, sensor readings, device logs, and cybersecurity events. The dataset comprised historical network traffic records, including information on communication protocols, packet-level details, and network flow data. Furthermore, the dataset included cybersecurity events and incidents recorded in the smart grid environment. These events encompassed a wide range of cybersecurity threats, including unauthorized access attempts, malware infections, network intrusions, and data breaches (Hussain et al., 2020). Using these datasets, the study of network behavior, spotting anomalies, and identifying potential cyber threats in the smart grid infrastructure can be done.

The data providers are examined to assess the sources and verify the integrity of the acquired information to ensure data quality and reliability. This included checking the data gathering methods, data storage protocols, and security measures in place to secure sensitive data. Various operational conditions, such as different periods, geographical locations, and system configurations, were considered to capture a wide range of cybersecurity events and activities. Overall, the dataset authenticity, representativeness, and comprehensiveness permitted their usage for conducting experiments that closely simulated real-world cybersecurity scenarios in the smart grid infrastructure. The employed datasets are Network Security Laboratory Knowledge Discovery in Databases (NSL-KDD Cup 99), the cyber security dataset of the Canadian Institute of Cybersecurity Intrusion Detection (CICIDS-2017), the power system attack detection dataset developed by the Oak Ridge national laboratory of Mississippi State University and the Washington University in St. Louis-Industrial

IoT-2018 and IoT 2021 (wustl-IIOT-2018, wustl-IIOT-2021) dataset for Industrial Control System SCADA cybersecurity.

4.2.2 Data cleaning and preprocessing

Before the analysis, the collected data underwent a series of preprocessing steps (Ibrahim et al., 2020) to ensure its quality and compatibility with machine learning algorithms. It included removing duplicate entries, handling missing values, normalization, standardization, and addressing any inconsistencies or outliers in the data.

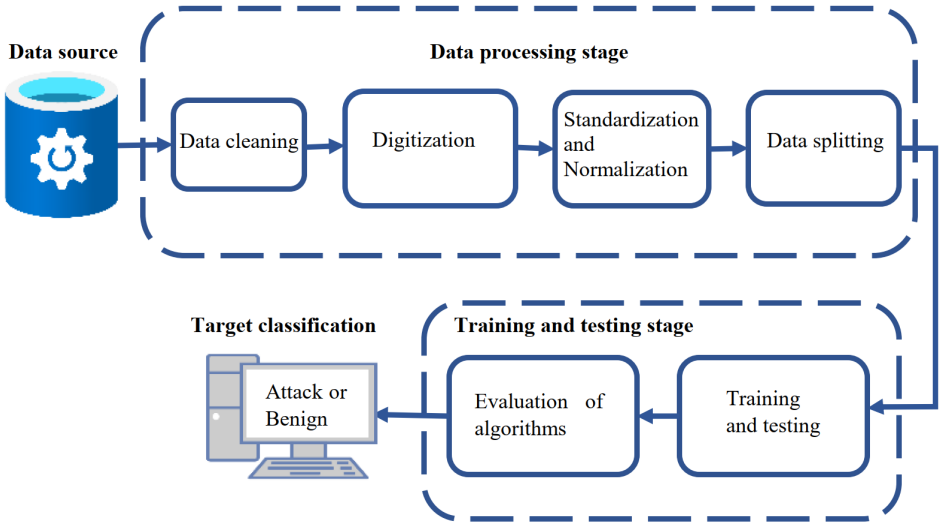


Figure 20: Data cleaning process.

4.2.3 Feature extraction

Feature extraction is crucial in data preprocessing, particularly in machine learning tasks. It involves transforming the raw data into representative features that can effectively capture the relevant information for the given problem or task. The goal is to derive a set of informative and discriminative features from the original data (Spataru, 2013) that could help differentiate between normal and malicious activities in the smart grid. This process involves applying various techniques to extract and represent the underlying patterns and structures in the data more meaningfully and compactly. There are several methods commonly used for feature extraction:

1. Dimensionality reduction techniques: These techniques aim to reduce the dimensionality of the data by transforming it into a lower-dimensional space

while preserving its essential characteristics. Examples include principal component analysis (PCA) and linear discriminant analysis.

2. Statistical methods: These methods involve computing statistical measures or descriptors from the data, such as mean, standard deviation, or histogram-based features. These features can provide insights into the distribution and variability of the data. They provide insights into the data distribution, central tendency, and dispersion, enabling the models to identify anomalies and abnormal behavior.

3. Transformations: Transforming the data using mathematical functions can help highlight specific patterns or relationships. For example, applying logarithmic, exponential, or Fourier transformations can reveal certain properties in the data.

In addition to these techniques, the domain-specific methods. The smart grid domain possesses unique characteristics and operational considerations that require specialized feature extraction approaches. Domain-specific features encompassed various aspects, including communication protocols, network topologies, operating parameters, and device configurations. Also, features derived from the physical parameters, such as voltage levels, current readings, and power consumption, can be extracted from sensor data with the domain-specific method. Feature extraction is performed in a way that balances the need for capturing relevant information with the goal of reducing dimensionality.

Dimensionality reduction techniques, such as restricted Boltzmann machines (RBM) and the genetically seeded flora (GSF) feature selection algorithms, were employed to mitigate the curse of dimensionality and remove redundant or irrelevant features. The extracted features formed the input variables for the machine learning models, enabling them to learn and make predictions based on the discriminative characteristics of the data. The models' ability to accurately classify and detect cybersecurity threats in the smart grid infrastructure is enhanced by extracting relevant features.

The feature extraction process involved employing statistical analysis techniques, signal processing methods, and domain-specific knowledge to extract meaningful and discriminative features from the raw data. The extracted features captured the statistical properties, temporal dependencies, and domain-specific indicators of cybersecurity threats in the smart grid. These features formed the input for the machine learning models, enabling them to effectively learn and make predictions in the context of securing the smart grid infrastructure.

From a different perspective, feature extraction can be seamlessly automated, eliminating the need for explicit data processing. This feat becomes achievable through the utilization of highly intricate deep learning algorithms, replete with several hidden layers. In such instances, time-series data has the potential to be transmuted

into 2D images or even into video signals. This approach has garnered significant acclaim for its ability to deliver heightened accuracy across a multitude of applications. Nonetheless, it's worth noting that the efficacy of this approach is contingent upon access to an extensive dataset, a resource that, unfortunately, is not easily accessible.

4.2.4 Feature selection techniques

With a potentially large number of features, feature selection techniques to identify the most informative and discriminative features for the analysis are employed. It involved evaluating each feature's correlation, significance, and relevance to identify and retain the most informative and relevant features that contribute significantly to the predictive performance of our models. This process helps to streamline the data, reducing dimensionality and mitigating the risk of overfitting, and ensuring that the features chosen capture the critical patterns and traits needed for reliable model training and evaluation.

Pearson correlation coefficient (PCC) analysis is carried out on the dataset to determine if it includes correlated features. PCC, or Pearson's r or simply correlation coefficient, is a statistical measure that quantifies the strength and direction of the linear relationship between two variables. Mathematically written as

$$\frac{\sum(x_i - x_j)(y_i - y_j)}{\sum\sqrt{(x_i - x_j)^2(y_i - y_j)^2}} \tag{56}$$

The symbol x_i represents the content of the variable in the dataset, while x_j refers to that variable's average value. Similarly, y_i represents the values of the sample y_j represents the average value of that variable. A correlation matrix is a square table listing each variable in rows and columns.

The diagonal elements of the matrix are always 1, indicating a perfect correlation of each variable with itself. The off-diagonal components represent the correlation coefficients between pairs of variables. Positive correlation coefficients close to 1 indicate that the variables tend to increase or decrease together, while negative correlation coefficients close to -1 indicate they move in opposite directions. Correlation matrices ranging from -1 to +1 are valuable for analyzing variable relationships, identifying data patterns, and discerning trends. Through thorough data preprocessing and feature selection, efforts are made to optimize the quality and representativeness of the input data, ultimately enhancing the overall performance and interpretability of the machine learning models in the subsequent stages of the study.

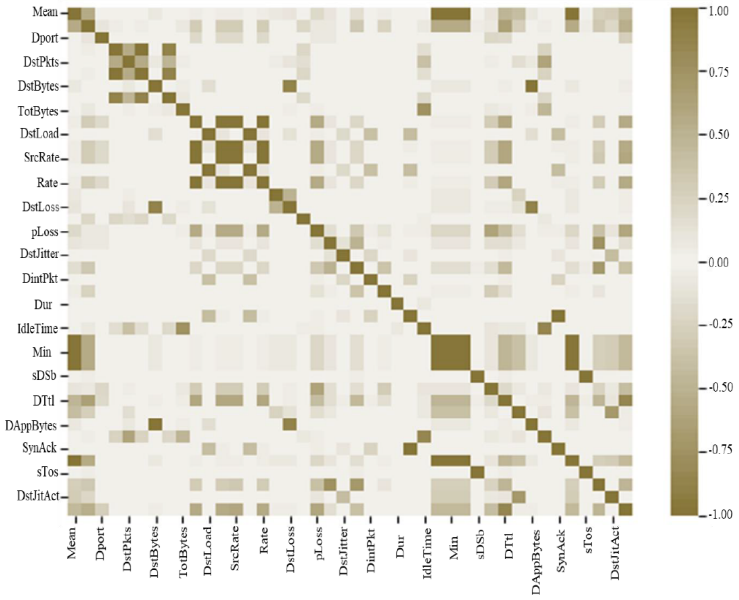


Figure 21: The correlation matrix for the wustle-2021 dataset.

4.3 Performance evaluation of machine learning algorithms

4.3.1 Evaluation metrics

In order to comprehensively assess the performance of the machine learning algorithms, a set of evaluation metrics are considered. The evaluation of the algorithm's categorization effectiveness is conducted using a confusion matrix, which serves as a method to assess the performance of a classifier algorithm. The confusion matrix is a fundamental scheme for evaluating the performance of machine learning models.

It consists of four essential parameters: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These parameters provide valuable insights into the model's predictions. TP represents the number of correct predictions of the positive class, TN represents the number of correct predictions of the negative class, FP indicates the number of incorrect predictions of the positive class, and FN signifies the number of incorrect predictions of the negative class. These metrics encompass a range of key performance indicators, including accuracy, precision, recall, and F1 score (Tang et al., 2023).

Accuracy: This is a frequently used metric in classification tasks, defined as the ratio of the number of correct predictions made by the model to the total number of predictions (Tang et al., 2023). Mathematically, the accuracy can be written as

$$Acuuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{57}$$

Precision: This metric is used to measure the precision of a classifier, and it is defined as the number of true positive predictions made by the model divided by the total number of positive predictions.

$$Precision = \frac{TP}{TP+FP} \tag{58}$$

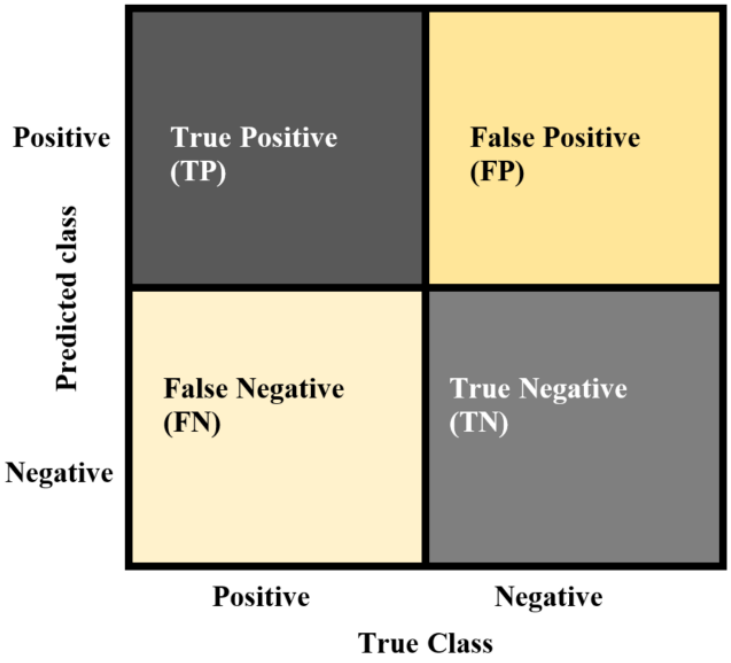


Figure 22: The confusion matrix.

Recall: This metric is used to measure the recall of a classifier, and it is defined as the number of true positive predictions made by the model divided by the number of positive cases in the dataset.

$$Recall = \frac{TP}{TP+FN} \tag{59}$$

F1 score: This is a metric that combines precision and recall, and it is defined as the

harmonic mean of precision and recall.

$$F1-score = \frac{2(Precision \times recall)}{Precision + recall} \tag{60}$$

These measures were considered because they provided essential insights into the model's performance in several areas. They made it possible to evaluate the model's overall performance, ability to classify cases accurately, capture pertinent information, and discriminatory power. With the help of a multi-dimensional evaluation strategy like this, it is possible to accurately assess the performance of the machine learning models and make decisions on their applicability and utility for the study.

To contextualize the presented confusion matrix within the realm of our cyber-security concerns in the smart grid, let's delve into a straightforward scenario. Imagine a situation where a remote sensor sends a message signaling a critical fault within the grid. The standard response to such a signal involves transmitting a protective message to trigger the opening of specific circuit breakers. This action aims to isolate the fault promptly, mitigating potential extensive and costly damages. However, it's imperative to recognize that the fault message from the sensor might be genuine or fabricated due to cyber-security attacks. This intricate determination falls under the purview of our proposed machine learning algorithm.

Consider that a positive output from the algorithm signifies the authenticity of the message, while a negative output suggests that the message is a result of a security breach. In this context, the evaluation pivots on comprehending the ramifications of both FP and FN. An FP arises when the machine learning algorithm inaccurately identifies a genuine fault, leading to the unnecessary activation of actions like switching the circuit breakers. The consequence of this scenario encompasses unwarranted energy loss and financial repercussions. Moreover, potential power outages and their associated adversities might ensue.

Conversely, an FN emerges when the algorithm erroneously classifies the message as fake, thus forgoing the activation of necessary actions even though a legitimate fault exists. This situation exposes the grid to substantial physical damages that could escalate to catastrophic levels in certain instances. This prompts an exploration into the critical theme of risk minimization. Risk minimization is one of the interesting future research topics our research team in Vaasa will handle.

4.3.2 Experimental setup

The dataset was meticulously divided into three sets to establish a robust experimental framework: training, validation, and testing. In some scenarios, the dataset is

usually divided into two subsets: training and testing datasets. Various approaches exist for partitioning datasets based on the percentages allocated for different purposes. Numerous studies have suggested configurations such as 80% for training and 20% for validation (Steimer, 2009), (Waqar et al., 2021), 70% for training and 30% for validation, and the emerging trend of 50% for training and 50% for validation. This partitioning yield effectively trains, fine-tunes, and assesses the performance of the machine-learning models.

To ensure the reliability and generalizability of the results, cross-validation techniques, which effectively mitigated potential biases and variance in the model performance are employed. The training set was the foundation for model training and hyperparameter tuning, optimizing the model's learning process and configuration. The validation set played a crucial role in model selection, providing an independent evaluation platform to identify the best-performing models based on their performance on unseen data. By employing this meticulous experimental setup, it became possible to minimize overfitting, choose models with superior performance, and attain a more comprehensive and trustworthy evaluation of the machine learning algorithms utilized.

4.3.3 Baseline models

Traditional cybersecurity detection methods commonly employed in the smart grid domain were implemented to establish a baseline for comparison. These methods included rule-based systems, anomaly detection algorithms, and signature-based approaches. The performance of the machine learning algorithms proposed in this study were evaluated against these baselines algorithms.

4.3.4 Comparison of machine learning algorithms

Experimentation was done with various machine learning algorithms, including decision trees, random forests, support vector machines (SVM), neural networks, and ensemble methods. For each algorithm, an evaluation is done with respect to their performance using the selected evaluation metrics. Statistical tests are conducted to identify significant differences between the models and determine the most effective approach.

4.4 Implementation of machine learning models

4.4.1 Model selection

The machine learning models' performance metrics and computational complexity are considered. The objective is to identify the most promising models that balance accuracy and efficiency, ensuring practical feasibility in real-world smart grid environments. The selection criteria incorporated multiple factors. The model's

performance metrics, such as accuracy, precision, recall, and F1 score, are examined. Models that consistently demonstrated high performance across these metrics, indicating their ability to detect and classify cybersecurity threats in the smart grid effectively, were sought.

Smart grid systems often require real-time or near-real-time analysis, making efficiency crucial. Therefore, the study assessed the computational requirements, including the training and inference times, memory utilization, and the complexity of the model architecture. Models that offered a good trade-off between performance and computational efficiency were prioritized. Consideration was also given to the scalability and adaptability of the models. The smart grid environment is dynamic, with new threats emerging and evolving. Models that could easily incorporate new data, adapt to changing conditions, and handle potential concept drift were preferred. This ensured the models'long-term applicability and sustainability. Furthermore, the feasibility of implementation was assessed. The selected models needed to be implementable within the existing smart grid infrastructure, taking into account the availability of resources, compatibility with the data collection and processing systems, and any specific constraints or limitations of the smart grid environment.

After careful evaluation and comparison, a set of machine learning models that fulfilled the aforementioned selection criteria were chosen for further analysis. These models formed the basis for the subsequent steps, including model training, hyperparameter tuning, and performance evaluation. A comprehensive evaluation of performance metrics, computational complexity, interpretability, scalability, adaptability, and feasibility drove the model selection process. By taking into account these factors, the objective was to identify machine learning models that show promise in enhancing the cybersecurity of the smart grid while also ensuring practical implementation and operational efficiency.

4.4.2 Model training

The selected machine learning models were trained using the training dataset carefully prepared during the data preprocessing stage. The training process involved utilizing appropriate training algorithms and optimization techniques to optimize the models' parameters and improve their performance. For each model, widely-used training algorithms were utilized, specifically tailored to the specific characteristics of the selected machine-learning technique. The choice of the training algorithm depended on the particular requirements and underlying principles of each machine learning model. The models iteratively adjusted their parameters during training to minimize the selected loss or error function. The optimization objective was to find the optimal parameter values that enable the models to accurately classify and detect cybersecurity threats in the smart grid.

4.4.3 Hyperparameter tuning

For the fine-tuning of the models, an extensive hyperparameter tuning process was conducted. This involved systematically exploring different combinations of hyperparameters and evaluating their impact on the models' performance. Grid and random search techniques were utilized to find the optimal hyperparameter values. Grid search involves exhaustively trying all possible combinations of hyperparameters within predefined ranges. It systematically covers the hyperparameter space and evaluates each combination's performance. On the other hand, random search randomly samples combinations from the hyperparameter space, which can be more efficient in certain scenarios. The performance of the models was assessed after each iteration of hyperparameter tuning. Hyperparameters are the configuration settings that control the behavior of the machine learning models, such as learning rates, regularization parameters, kernel choices, or the number of hidden layers. Hyperparameter tuning is essential to fine-tune the models and find the optimal combination of hyperparameter values. This process involved systematically exploring different combinations of hyperparameters and evaluating their impact on the models' performance.

Cross-validation techniques were employed to assess the models' performance across various hyperparameter settings. Cross-validation partitions the training dataset into multiple subsets, with one subset used as a training set while the rest are used for testing or validation. Once the models have been trained, evaluating their performance using validation and testing datasets is essential. The validation dataset is used during training to assess the model's performance on data it has not seen before. This helps monitor the model's progress and decide when to stop training to prevent overfitting. The testing dataset, which is independent of the training and validation data, is used as a final assessment of the model's performance. It objectively evaluates how well the model generalizes to unseen data. This enabled determining the hyperparameter settings that result in the best generalization and estimating the models' performance on hypothetical data. The model was trained using the training dataset for each combination of hyperparameters, and its performance was evaluated using the validation dataset. Performance metrics were computed to assess the effectiveness of the models under different hyperparameter settings.

5 RESULTS AND DISCUSSION

5.1 Introduction

The preceding chapter presented a theoretical framework for detecting cyber-attacks on the smart grid infrastructure. The framework established a comprehensive system model through mathematical deductions. This section delves into the practical aspects of identifying cyber-attacks embedded in various smart grids using machine learning algorithms. The situations involved in this process and detailed simulations to demonstrate the effectiveness of the algorithms are described. The analysis findings, offering valuable insights into the performance and accuracy of the detection system, are accompanied by discussions of the implications and interpretations derived from the results.

5.2 Simulation results and discussion

In this section, the performance of various machine learning algorithms is evaluated through simulations. The simulations in this work were conducted using two software platforms: MATLAB version R2021a and Jupyter Notebook. MATLAB is a high-level programming language and interactive environment developed by MathWorks. It provides a wide range of mathematical and scientific functions for numerical computation, data visualization, algorithm development, and application development. MATLAB enables matrix computations, linear algebra, optimization, data analysis, and signal processing. It offers powerful tools for visualizing data through various types of plots and supports the development of algorithms for mathematical models, machine learning, image processing, and more. MATLAB also allows for the creation of standalone applications and user interfaces. Additionally, it supports integration with other programming languages, enabling leveraging existing code or using MATLAB functions within other environments.

On the other hand, Jupyter Notebook is another robust platform known for its strength in handling machine learning algorithms. It is an open-source web application that provides an interactive computing environment for creating and sharing documents containing live code, equations, visualizations, and explanatory text. With its interactive and flexible nature, Jupyter Notebook provides an ideal environment for developing and implementing machine learning models. By leveraging the capabilities of both MATLAB and Jupyter Notebook, this study explores and compares the performance of different machine learning algorithms. This combination of software platforms ensures a comprehensive analysis and allows for a thorough evaluation of the algorithms under consideration. The utilization of MATLAB and Jupyter Notebook empowers this research to effectively investigate and assess the performance of machine learning algorithms, ultimately contributing to a comprehensive understanding of their effectiveness in the context of this study.

5.3 Model evaluation

After training and hyperparameter tuning, the selected models were evaluated. The evaluation aimed to measure the models' performance on unseen data and assess their generalization capability. The evaluation metrics discussed earlier, such as accuracy, precision, recall, and F1 score, were computed to gauge the models' performance (Berghout & Benbouzid, 2022). These metrics provided insights into the models'ability to correctly classify cybersecurity threats in the smart grid and their overall effectiveness in enhancing cybersecurity. The evaluation results were used to compare the performance of different models, identify the best-performing model, and provide intuitions about the models strengths and weaknesses.

In that sense is publication II, where the proposed algorithm, hybridized by the convolutional neural network (CNN) and the gated recurrent unit (GRU) performance, is compared to other deep learning methods, such as CNN, GRU, and long short-term memory (LSTM) to detect cyber-attacks in smart grids. Using the hyperparameters depicted in Table 1 and employing the CICIDS2017 dataset for experimentation, the compared algorithms for intrusion detection are evaluated using the standard evaluation metrics.

Number	Parametric	Quantity
1	Input layer	78
2	Hidden layer	55
3	Activation function	ReLU
4	Iteration limit	1000
5	Cost function	Cross entropy
6	Batch size	128

Table 1: The configuration of the hyperparameters.

Figure 23 illustrates the overall performance comparison of the considered algorithms as presented in publication II. These findings underscore the effectiveness and potential of the proposed algorithm in accurately detecting and classifying the target. The achieved accuracy of 99.7% demonstrates a significant improvement over the existing techniques, highlighting the superiority of the proposed algorithm. By achieving such high accuracy, the proposed algorithm showcases its ability to handle the complexities of the task effectively. This result holds considerable promise for practical applications where accurate detection is paramount.

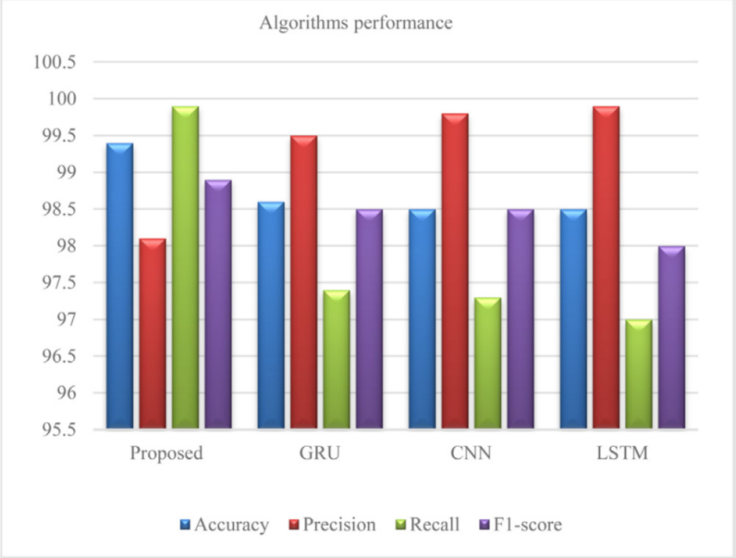


Figure 23: Overall performance comparison of the considered algorithms.

5.4 Sensitivity analysis

Sensitivity analysis examines how modifications to a model's input parameters impact the model's output. It is a method of figuring out how sensitive a model's output is to changes in its inputs (Saltelli et al., 2010; Shin et al., 2013; Ballester-Ripoll et al., 2019; Guo et al., 2011; Wagener & Pianosi, 2019). It can be used to answer questions such as: How much will the model's output change with a change of a certain value of the input variable? Which input parameters have the most significant influence on the model's output? Given the ambiguity in the input variables, what range of values can the model's output assume? There are two main types of sensitivity analysis: One-factor sensitivity analysis: This type of analysis examines the impact of changes in a single input variable on the model's output. Multi-factor sensitivity analysis: This type of analysis examines the effect of changes in multiple input variables on the model's output.

A sensitivity analysis was conducted to assess the employed machine learning algorithms' robustness. The study evaluated the models' performance under various scenarios, such as introducing noise in the data, simulating adversarial attacks, or altering the data distribution. The purpose is to gain insights into the behavior and performance of the algorithms under different conditions, helping to:

1. Identify critical factors: By varying one input parameter at a time while keeping others constant, sensitivity analysis helps identify which factors influence the output most. These critical factors are essential for understanding the model's behavior and can aid in focusing resources on the most impactful areas.

2. Assess robustness and reliability: Sensitivity analysis is valuable in evaluating how robust and reliable a model or system is when faced with uncertainties or changes in input parameters. It provides an understanding of the system's output sensitivity to fluctuations in its inputs, offering insights into the stability and accuracy of the model.

3. Optimize decision-making: Sensitivity analysis can guide decision-making processes by determining the range of input values within which the model performs satisfactorily. This helps in identifying the optimal conditions or parameters for achieving desired outcomes.

4. Risk assessment: By evaluating the effects of different inputs on the model's output, sensitivity analysis aids in assessing potential risks and uncertainties associated with the model or process. It highlights areas where uncertainties may have the most substantial impact on outcomes, enabling risk mitigation strategies to be developed.

5. Model validation: Sensitivity analysis is critical in model validation and verification. It helps researchers and analysts ensure that their models are not overly sensitive to specific inputs or assumptions and provide reasonable and consistent results.

Sensitivity analysis can take various forms, depending on the system's complexity. Some standard techniques include:

- One-at-a-time (OAT) sensitivity analysis: This involves varying one input parameter at a time while keeping all others constant and observing the resulting changes in the output (Federico Ferretti , Andrea Saltelli, 2016).

- Local sensitivity analysis: Local sensitivity analysis (Federico Ferretti , Andrea Saltelli, 2016) involves making small perturbations to the input parameters around their current values and observing how these changes affect the model's output. This permits assessing the model's sensitivity to variations in the input parameters in the immediate vicinity of their current settings.

- Global sensitivity analysis: This method explores the entire input parameter space to determine the collective impact of multiple inputs on the output. It helps understand how uncertainties in different input parameters interact and influence the system's behavior.

Sensitivity analysis is widely used in various fields, including engineering, finance, environmental sciences, and decision analysis, where understanding the effects of uncertain factors is crucial for making informed decisions and optimizing performance. This was considered in publication V, where the WUSTL_IIoT_2021 dataset and the WUSTL_IIoT_2018 dataset used for analysis were manipulated to reflect this notion. The performance of some leading algorithms was evaluated, and the results are shown in Figures 24 and Figure 25, respectively.

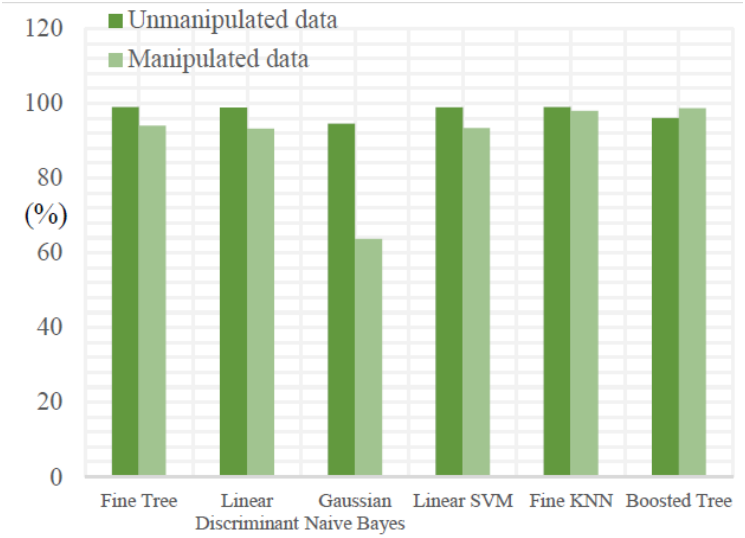


Figure 24: Performance of the best-performing algorithms on the WUSTL_IIoT_2018 dataset.

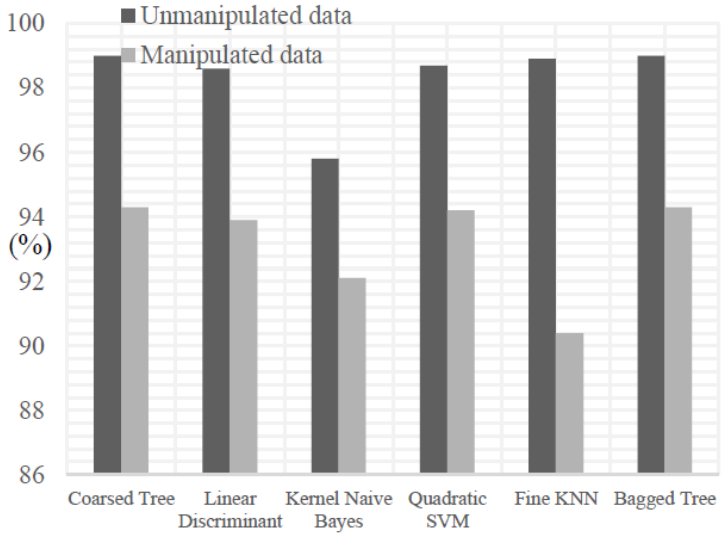


Figure 25: Performance of the best-performing algorithms on the WUSTL_IIoT_2021 dataset.

The accuracy of the fine tree, linear discriminant, and linear SVM algorithms slightly declined when trained on manipulated data, as shown in Figure 24. The fine KNN algorithm showed little degradation, while the Gaussian Naive Bayes algorithm experienced a significant decline in accuracy. Surprisingly, the booted tree algorithm's accuracy increased from 96.1% with un-manipulated data to 98% with manipulated

data. Figure 24 shows the best-performing algorithms' performance analysis. When tested on manipulated data, the coarse tree, linear discriminant, quadratic SVM, and bagged tree classifiers experienced a moderate decrease in their performance. On the other hand, the fine KNN classifier showed a significant drop in accuracy, while the Kernel Naive Bayes classifier performed poorly on both un-manipulated and manipulated data. The results of these experiments suggest that the accuracy of machine learning algorithms can be affected by the quality of the training data.

In this case, the manipulated data caused the accuracy of some algorithms to decline, while the accuracy of other algorithms increased. This suggests that the impact of data manipulation on machine learning algorithms can vary depending on the specific algorithm and the type of manipulation performed. The experimental outcomes shed light on the critical influence of training data quality on the accuracy of machine learning algorithms. It becomes evident that the performance of these algorithms can be significantly affected by the nature of the data they are trained on. Interestingly, the effect of data manipulation was not uniform across all algorithms; instead, it led to varying outcomes. Specifically, the experiment revealed that some algorithms experienced a decline in accuracy when confronted with manipulated data. On the other hand, some algorithms improved accuracy under the same conditions. These findings emphasize the intricate connection between data manipulation and algorithm behavior. It demonstrates that the impact of data manipulation on algorithms is complex and multifaceted. Different algorithms respond differently to data alterations, leading to varied outcomes in terms of their performance. This emphasizes the need to understand how each algorithm behaves under different data conditions thoroughly.

5.5 Risk analysis

In the context of applying machine learning algorithms to classification problems, risk analysis is the process of identifying, assessing, and managing potential risks associated with the model's predictions and performance. Risk analysis plays a crucial role in applying machine learning algorithms as it aims to identify and understand the machine learning algorithm's uncertainties, limitations, and possible drawbacks, mainly when deployed in real-world applications. When utilizing machine learning algorithms for classification tasks, some common risk factors to consider include:

- Accuracy and performance risks: Assessing the accuracy and performance of the classification model is crucial. Understanding its strengths and limitations in different scenarios is essential to avoid incorrect predictions and costly errors.

- False positives and false negatives: In binary classification problems, false positives and false negatives can have varying consequences depending on

the application. Risk analysis helps understand the implications of these misclassifications and how they might impact decision-making.

- Data quality and bias: Machine learning models are highly dependent on the quality and representativeness of the training data. If that data is biased or contains inaccuracies, the algorithm may reproduce those biases or inaccuracies in its predictions. It is essential to thoroughly analyze the dataset to identify any existing biases or inaccuracies and address them appropriately before applying the algorithm. Risk analysis involves investigating potential biases in the data and understanding how they might lead to biased or unfair predictions.

- Robustness to adversarial attacks: The model's vulnerability to adversarial attacks should be evaluated for specific applications, especially in security-sensitive areas. Adversarial attacks intentionally perturb the input data to cause misclassifications, which can be a significant risk in critical applications.

- Generalization and overfitting: The concept of generalization and overfitting is crucial in assessing the model's ability to perform well on new, unseen data. Generalization refers to how effectively the model can extend its learned patterns to new instances, ensuring its robustness and reliability. On the other hand, overfitting occurs when the model becomes excessively tailored to the training data, losing its ability to generalize accurately to new data. Consequently, an overfitted model may yield poor performance and unreliable predictions when applied in real-world situations. Thus, it is essential to strike a balance during model training to avoid overfitting and achieve better generalization on unseen data.

- Interpretability and explainability: Understanding how the machine learning model arrives at its predictions for some applications is crucial. Risk analysis may involve assessing the interpretability and explainability of the model to ensure it can be trusted and understood by stakeholders. Some machine learning algorithms, such as deep neural networks, are known for their black-box nature, meaning it can be challenging to understand why they make specific predictions or decisions. This lack of interpretability can be problematic, especially in sensitive domains like smart grids, healthcare, or finance.

- Model drift: The data distribution in real-world applications may change over time, leading to model drift. Risk analysis involves monitoring the model's performance over time and taking measures to mitigate the effects of drift.

- Legal and ethical considerations: Deploying machine learning models for classification tasks may raise legal and ethical issues, particularly when dealing with sensitive data or making decisions with significant consequences.

Risk analysis includes identifying potential legal or ethical implications and ensuring compliance with relevant regulations.

Conducting risk analysis involves performing sensitivity analyses, employing performance metrics that consider misclassification consequences, evaluating model robustness under various conditions, and conducting comprehensive testing in diverse real-world scenarios. By conducting a thorough risk analysis, developers and stakeholders can make informed decisions about deploying machine learning models, manage potential risks, and enhance the trustworthiness and reliability of the system in real-world applications.

6 CONCLUSION

6.1 Conclusion

This doctoral dissertation thoroughly examines machine learning techniques to improve cybersecurity in the smart grid. The smart grid holds enormous promise for effective and sustainable energy management due to its integration of digital technology and communication networks. However, this digital transition also brings cybersecurity issues, peaking the creation of solid safeguards for sensitive data and this critical infrastructure. To address these challenges, we delved into the realm of machine learning and its role in bolstering smart grid cybersecurity. Various machine learning algorithms were explored, evaluating their suitability for cybersecurity applications in the smart grid context. This understanding of machine learning's potential and limitations positioned the dissertation to create innovative solutions. Building upon this foundation, comprehensive methodologies were proposed for applying machine learning to the cybersecurity of the smart grid.
Specifically:

a) We proposed a machine learning algorithm for an intrusion detection system (IDS) in the supervisory control and data acquisition (SCADA) system applied in the smart grid. We found that the proposed algorithm achieves a detection accuracy much better than existing traditional IDSs for SCADA systems. This is shown in paper I.

b) We propose a hybrid deep learning algorithm that focuses on distributed denial of service (DDoS) attacks on the communication infrastructure of the smart grid. The convolutional neural network (CNN) and the gated recurrent unit (GRU) algorithms hybridize the proposed algorithm. The proposed algorithm outperforms the counter-algorithms in terms of overall accuracy. This is shown in paper II.

c) We evaluated the performance of traditional supervised machine learning algorithms like artificial neural networks (ANN), CNN, and support vector machines (SVM) against a proposed Restricted Boltzmann Machine-based nature-inspired artificial root foraging optimization algorithm. The proposed algorithm yielded better results in comparison. This is shown in paper III.

d) A genetically seeded flora transformer neural network (GSFTNN) algorithm stands in stark contrast to the signature-based method employed by traditional IDSs was proposed. The proposed algorithm outperforms traditional algorithms such as residual neural networks (ResNet), recurrent neural networks (RNN), and long short-term memory (LSTM) in terms of accuracy and efficiency. This is shown in paper IV.

e) We evaluated the performance of some machine learning algorithms by manually introducing adversarial attacks on the dataset. We determined that train-

ing machine learning algorithms with ill-nature data could affect the algorithm's performance. This is shown in paper V.

Drawing from our research endeavors, it becomes evident that AI and machine learning possess the capacity to elevate the realm of cybersecurity tools to unprecedented heights. Our exploration has encompassed a selection of algorithms, yet the landscape abounds with many untapped options with the potential for superior performance. Take, for instance, reinforcement learning - an emerging contender. However, the successful integration of reinforcement learning mandates deployment within a dynamic simulation environment, one replete with authentic data that mirrors reality. This setting facilitates the crucial self-learning process through iterative massive trial and error, guided by the feedback of rewards and penalties.

6.2 Future research

These future research directions build upon the significant findings and contributions of the Ph.D. dissertation, offering exciting opportunities to advance the fields of machine learning and cybersecurity for the smart grid. By addressing these challenges and exploring novel approaches, researchers can contribute to creating more resilient, secure, and efficient smart grid systems that can better withstand the ever-evolving landscape of cyber threats.

a) Adversarial robustness and resilience: Adversarial attacks pose significant threats to machine learning models in the smart grid. Future research should explore methods to improve the robustness and resilience of machine learning algorithms against various adversarial attacks, including those specifically tailored to target smart grid applications. This research could involve developing novel defense mechanisms, data augmentation techniques, and advanced training methodologies to mitigate the impact of adversarial manipulations.

b) Real-time cybersecurity monitoring and response: The smart grid operates in real-time, making it essential to have cybersecurity solutions that can respond quickly to emerging threats. Future research can focus on developing real-time cybersecurity monitoring and response systems that leverage machine learning for rapid threat detection and adaptive countermeasures. These systems should be able to detect anomalies, identify cyberattacks, and autonomously implement defensive actions to protect the smart grid infrastructure.

c) Blockchain for smart grid security: Blockchain technology offers the potential to enhance the security and integrity of smart grid data and communication. Future research could investigate how machine learning algorithms can be combined with blockchain to create secure and tamper-resistant smart grid applications. This includes exploring the use of blockchain for secure data storage, provenance tracking, and decentralized cybersecurity decision-making.

d) Multi-modal data fusion for enhanced security: Combining data from various sources, such as smart meters, IoT devices, and weather sensors, can provide a more comprehensive understanding of the smart grid's security posture. Future research could investigate multi-modal data fusion techniques using machine learning to improve situational awareness and early threat detection.

e) Cyber-physical security integration: The smart grid is a cyber-physical system, and future research could explore integrating machine learning techniques with physical security measures. This could involve using sensor data, video analytics, and machine learning algorithms to detect physical intrusions or tampering attempts that might compromise the smart grid's security.

f) Human-centric cybersecurity: In the smart grid context, human operators and users play a crucial role in ensuring cybersecurity. Future research can focus on incorporating human-centric approaches into machine learning-based cybersecurity systems. This involves considering the cognitive aspects of human decision-making, usability, and user-centric design principles to create cybersecurity solutions that are intuitive and easy for human operators to interact with effectively.

g) Hardware-supported cybersecurity: Research can explore the integration of machine learning algorithms with hardware-level security features to enhance the resilience of smart grid systems. Hardware-based security mechanisms, such as hardware-enforced isolation and trusted execution environments, can complement machine learning algorithms to provide additional protection against attacks.

REFERENCES

Anwar, A., & Mahmood, A. N. (2014). Cyber security of smart grid infrastructure. arXiv preprint arXiv:1401.3936.

Ahmed, S. D., Al-Ismail, F. S. M., Shafiullah, M., Al-Sulaiman, F. A., & El-Amin, I. M. (2020). Grid Integration Challenges of Wind Energy: A Review. IEEE Access, 8(type 1), 10857-10878.https://doi.org/10.1109/ACCESS.2020.2964896

Al Ameen, M., Liu, J., & Kwak, K. (2012). Security and privacy issues in wireless sensor networks for healthcare applications. Journal of Medical Systems, 36(1), 93-101. https://doi.org/10.1007/s10916-010-9449-4.

Alkuwari, A. N., Al-Kuwari, S., & Qaraqe, M. (2022). Anomaly Detection in Smart Grids: A Survey from Cybersecurity Perspective*. 3rd International Conference on Smart Grid and Renewable Energy, SGRE 2022 - Proceedings, 1-7. https://doi.org/10.1109/SGRE53517.2022.9774221

Anish Halimaa, K. S. (2019). Machine Learning Based Intrusion Detection System.

Arabo, A. (2015). Cyber Security Challenges within the Connected Home Ecosystem Futures. Procedia Computer Science, 61(0), 227-232.
https: doi.org/10.1016/j.procs.2015.09.201

Ashok, A., Govindarasu, M., & Wang, J. (2017). Cyber-Physical Attack-Resilient Wide-Area Monitoring, Protection, and Control for the PowerGrid. 1-17.

Avancini, D. B., Martins, S. G. B., Rabelo, R. A. L., Solic, P., & Rodrigues, J. J. P. C. (2018). A Flexible IoT Energy Monitoring Solution. 2018 3rd International Conference on Smart and Sustainable Technologies, SpliTech 2018, 1-6.

Ayar, M., Obuz, S., Trevizan, R. D., Bretas, A. S., & Latchman, H. A. (2017). A Distributed Control Approach for Enhancing Smart Grid Transient Stability and Resilience. IEEE Transactions on Smart Grid, 8(6), 3035-3044.
https://doi.org/10.1109/TSG.2017.2714982

Azad, S., Sabrina, F., & Wasimi, S. (2019). Transformation of smart grid using machine learning. 2019 29th Australasian Universities Power Engineering Conference,
AUPEC 2019. https://doi.org/10.1109/AUPEC48547.2019.211809

Baig, Z. A., & Amoudi, A. R. (2013). An analysis of smart grid attacks and coun-

termeasures. *Journal of Communications*, 8(8), 473-479. https://doi.org/10.12720/jcm.8.8.473-479

Basnet, M., Poudyal, S., Ali, M. H., & Dasgupta, D. (2021). Ransomware detection using deep learning in the SCADA system of electric vehicle charging station. 2021 IEEE PES Innovative Smart Grid Technologies Conference - Latin America, ISGT Latin America 2021, 1-5. https://doi.org/10.1109/ISGTLatinAmerica52371.2021.9543031

Batta, M. (2018). Machine Learning Algorithms - A Review. International Journal of Science and Research (IJSR), 18(8), 381-386. https://doi.org/10.21275/ART20203995

Belgiu, M., & Dragu, L. (2016). Random forest in remote sensing: A review of applications and future directions. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24-31. https://doi.org/10.1016/j.isprsjprs.2016.01.011

Belkebir, N., Maaroufi, M., Khallaayoun, A., & Lghoul, R. (2018). The Future Development of Smart Grid The case of Morocco. c, 1-9.

Berghout, T., & Benbouzid, M. (2022). EL-NAHL: Exploring labels autoencoding in augmented hidden layers of feedforward neural networks for cybersecurity in smart grids. Reliability Engineering and System Safety, 226(January), 108680. https://doi.org/10.1016/j.ress.2022.108680

Bhattarai, B. P., Paudyal, S., Luo, Y., Mohanpurkar, M., Cheung, K., Tonkoski, R., Hovsapian, R., Myers, K. S., Zhang, R., Zhao, P., Manic, M., Zhang, S., & Zhang, X. (2019). Big data analytics in smart grids: State-of-theart, challenges, opportunities, and future directions. IET Smart Grid, 2(2), 141-154. https://doi.org/10.1049/iet-stg.2018.0261

Blumsack, S., & Fernandez, A. (2012). Ready or not, here comes the smart grid! Energy, 37(1), 61-68. https://doi.org/10.1016/j.energy.2011.07.054

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys and Tutorials, 18(2), 1153-1176. https://doi.org/10.1109/COMST.2015.2494502

Butun, I., Sari, A., & Osterberg, P. (2019). Security Implications of Fog Computing on the Internet of Things. 2019 IEEE International Conference on Consumer Electronics, ICCE 2019, 20201010, 1?6. https://doi.org/10.1109/ICCE.2019.8661909

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends, 2(01), 20-28. https://doi.org/10.38094/jastt20165

Chicco, G., Riaz, S., Mazza, A., & Mancarella, P. (2020). Flexibility from Distributed Multienergy Systems. Proceedings of the IEEE, 108(9), 1496-1517. https://doi.org/10.1109/JPROC.2020.2986378

Cone, B. D., Irvine, C. E., Thompson, M. F., & Nguyen, T. D. (2007). A video game for cyber security training and awareness. Computers and Security, 26(1), 63-72. https://doi.org/10.1016/j.cose.2006.10.005

de Souza, C. A., Westphall, C. B., Machado, R. B., Loffi, L., Westphall, C. M., & Geronimo, G. A. (2022). Intrusion detection and prevention in fog based IoT environments: A systematic literature review. Computer Networks, 214(July), 109154. https://doi.org/10.1016/j.comnet.2022.109154

Deb, S., Vatwani, T., Chattopadhyay, A., Basu, A., & Fong, X. (2018). Domain Wall Motion-Based Dual-Threshold Activation Unit for Low-Power Classification of Non-Linearly Separable Functions. IEEE Transactions on Biomedical Circuits and Systems, 12(6), 1410-1421. https://doi.org/10.1109/TBCAS.2018.2867038

Dehalwar, V., Kalam, A., & Zayegh, A. (2014). Infrastructure for real-time communication in smart grid. 2014 Saudi Arabia Smart Grid Conference, SASG 2014, 2-5. https://doi.org/10.1109/SASG.2014.7274281

Diaba, S. Y., Shafie-khah, M., & Elmusrati, M. (2022). On the performance metrics for cyber-physical attack detection in smart grid. Soft Computing, 26(23), 13109?13118. https://doi.org/10.1007/s00500-022-06761-1

Diamantoulakis, P. D., Kapinas, V. M., & Karagiannidis, G. K. (2015). Big Data Analytics for Dynamic Energy Management in Smart Grids. Big Data Research, 2(3), 94-101. https://doi.org/10.1016/j.bdr.2015.03.003

Ding, J., Qammar, A., Zhang, Z., Karim, A., & Ning, H. (2022). Cyber Threats to Smart Grids: Review, Taxonomy, Potential Solutions, and Future Directions. Energies, 15(18), 1-37. https://doi.org/10.3390/en15186799

Diptiben Ghelani, (2022). Cyber Security in Smart Grids, Threats, and Possible Solutions. American Journal of Applied Scientific Research, Vol. x(x.2022).

Dogaru, D. I., & Dumitrache, I. (2019). Cyber security of smart grids in the context

of big data and machine learning. Proceedings - 2019 22nd International Conference on Control Systems and Computer Science, CSCS 2019, 61-67. https://doi.org/10.1109/CSCS.2019.00018

Dong, H., Supratak, A., Pan, W., Wu, C., Matthews, P. M., & Guo, Y. (2018). Mixed Neural Network Approach for Temporal Sleep Stage Classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26(2), 324-333. https://doi.org/10.1109/TNSRE.2017.2733220

Eder-Neuhauser, P., Zseby, T., Fabini, J., & Vormayr, G. (2017). Cyber attack models for smart grid environments. Sustainable Energy, Grids and Networks, 12, 10-29. https://doi.org/10.1016/j.segan.2017.08.002

Eken, H. (2013). Security Threats and Solutions in Cloud Computing. World Congress on Internet Security (WorldCIS-2013).

Espe, E., Potdar, V., & Chang, E. (2018). Prosumer communities and relationships in smart grids: A literature review, evolution and future directions. Energies, 11(10). https://doi.org/10.3390/en11102528

Falahi, M., Vasilateanu, A., Goga, N., Suciu, G., Sachian, M. A., Florescu, R., Ali, H. A., & Qian, Y. (2022). An Innovative Blockchain System for Smart Grids. 2022 IEEE International Conference on Blockchain, Smart Healthcare and Emerging Technologies, SmartBlock4Health 2022. https://doi.org/10.1109/SmartBlock4Health56071.2022.10034523

Fang, X., Misra, S., Xue, G., & Yang, D. (2012). Smart grid - The new and improved power grid: A survey. IEEE Communications Surveys and Tutorials, 14(4), 944-980. https://doi.org/10.1109/SURV.2011.101911.00087

Federico Ferretti , Andrea Saltelli, S. T. (2016). Trends in sensitivity analysis practice in the last decade.Science of The Total Environment, 568, 666-670. https://doi.org/10.1016/j.scitotenv.2016.02.133

Gao, L., Pi, Z., Wang, J., & Sun, J. (2022). Smart Grid Data Traceability System based on Blockchain Technologies. Proceedings - 2022 8th Annual International Conference on Network and Information Systems for Computers, ICNISC 2022, 721-726. https://doi.org/10.1109/ICNISC57059.2022.00145

Gunduz, M. Z., & Das, R. (2020). Cyber-security on smart grid: Threats and potential solutions. Computer Networks, 169, 107094. https://doi.org/10.1016/j.comnet.2019.107094

Hahn, A., Ashok, A., Sridhar, S., & Govindarasu, M. (2013). Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid. IEEE Transactions on Smart Grid, 4(2), 847-855. https://doi.org/10.1109/TSG.2012.2226919

Haluk Gozde; M. Cengiz Taplamacioglu; Murat Ari; Hamza Shalaf. (2015). 4G/LTE technology for smart grid communication infrastructure. 2015 *3rd International Istanbul Smart Grid Congress and Fair (ICSG)*.

He, H., & Yan, J. (2016). Cyber-physical attacks and defences in the smart grid: a survey. IET Cyber-Physical Systems: Theory & Applications, 1(1), 13-27. https://doi.org/10.1049/iet-cps.2016.0019

Hledik, R. (2009). How Green Is the Smart Grid? Electricity Journal, 22(3), 29-41. https://doi.org/10.1016/j.tej.2009.03.001

Huang, B. B., Xie, G. H., Kong, W. Z., & Li, Q. H. (2012). Study on smart grid and key technology system to promote the development of distributed generation. 2012 IEEE Innovative Smart Grid Technologies - Asia, ISGT Asia 2012, 1-4. https://doi.org/10.1109/ISGT-Asia.2012.6303265

Hussain, H. M., Narayanan, A., Nardelli, P. H. J., & Yang, Y. (2020). What is energy internet? concepts, technologies, and future directions. IEEE Access, 8(iv), 183127-183145. https://doi.org/10.1109/ACCESS.2020.3029251

Ibrahim, M. S., Dong, W., & Yang, Q. (2020). Machine learning driven smart electric power systems: Current trends and new perspectives. Applied Energy, 272(June), 115237. https://doi.org/10.1016/j.apenergy.2020.115237

ICS-CERT. (2015). ICS-CERT Monitor September 2014 - Feburary 2015. ICS-CERT Monitor, February, 1?15.

Jamil, F., Iqbal, N., Imran, Ahmad, S., & Kim, D. (2021). Peer-to-Peer Energy Trading Mechanism Based on Blockchain and Machine Learning for Sustainable Electrical Power Supply in Smart Grid. IEEE Access, 9, 39193-39217. https://doi.org/10.1109/ACCESS.2021.3060457

Javid, I., Ghazali, R., Syed, I., Husaini, N. A., & Zulqarnain, M. (2022). Developing Novel T-Swish Activation Function in Deep Learning. 2022 International Conference on IT and Industrial Technologies, ICIT 2022, 1-7. https://doi.org/10.1109/ICIT56493.2022.9989151

Jozwiak, D., Pillai, J. R., Ponnaganti, P., Bak-Jensen, B., Jantzen, J., Wu, X., Dai,

H., Zhang, N., Kong, W., O'Malley, M. J., Anwar, M. B., Heinen, S., Kober, T., McCalley, J., McPherson, M., Muratori, M., Orths, A., Ruth, M., Schmidt, T. J., Maniatakos, M. (2020). Electric Power Grid Resilience to Cyber Adversaries: State of the Art. IEEE Access, 8(5), 4929-4934. https://doi.org/10.1109/TII.2021.3112095

Kappagantu, R., & Daniel, S. A. (2018). Challenges and issues of smart grid implementation: A case of Indian scenario. Journal of Electrical Systems and Information Technology, 5(3), 453-467.
https://doi.org/10.1016/j.jesit.2018.01.002

Kimani, K., Oduol, V., & Langat, K. (2019). Cyber security challenges for IoT-based smart grid networks. *International Journal of Critical Infrastructure Protection*, 25, 36-49.
https://doi.org/10.1016/j.ijcip.2019.01.001

Kulkarni, V. Y., & Sinha, P. K. (2012). Pruning of random forest classifiers: A survey and future directions. Proceedings-2012 International Conference on Data Science and Engineering, ICDSE 2012, 64-68. https://doi.org/10.1109/ICDSE.2012.6282329

Kumar, D., & S., D. S. (2020). Enhancing Security Mechanisms for Healthcare Informatics Using Ubiquitous Cloud. Journal of Ubiquitous Computing and Communication Technologies, 2(1), 19-28.
https://doi.org/10.36548/jucct.2020.1.003

Kumar, V., Pandey, A. S., & Sinha, S. K. (2016). Grid integration and power quality issues of wind and solar energy system: A review. International Conference on Emerging Trends in Electrical, Electronics and Sustainable Energy Systems, ICETEESES 2016, 2011, 71-80.
https://doi.org/10.1109/ICETEESES.2016.7581355

Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M. Y. Bin, Bamhdi, A. M., & Budiarto, R. (2020). CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection. IEEE Access, 8, 132911-132921. https://doi.org/10.1109/ACCESS.2020.3009843

Kuzlu, M., Sarp, S., Pipattanasomporn, M., & Cali, U. (2020). Realizing the potential of blockchain technology in smart grid applications. 2020 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2020, 1-5. https://doi.org/10.1109/ISGT45199.2020.9087677

Lamba, V., Simkova, N., & Rossi, B. (2019). Recommendations for smart grid security risk management. Cyber-Physical Systems, 5(2), 92-118.
https://doi.org/10.1080/23335777.2019.1600035

Leszczyna, R. (2018a). Cybersecurity and privacy in standards for smart grids - A comprehensive survey. Computer Standards and Interfaces, 56(September 2017), 62-73. https://doi.org/10.1016/j.csi.2017.09.005

Leszczyna, R. (2018b). Standards on cyber security assessment of smart grid. International Journal of Critical Infrastructure Protection, 22, 70-89.
https://doi.org/10.1016/j.ijcip.2018.05.006

Li, T., Zhang, W., Chen, N., Qian, M., & Xu, Y. (2019). Blockchain Technology Based Decentralized Energy Trading for Multiple-Microgrid Systems. 2019 3rd IEEE Conference on Energy Internet and Energy System Integration: Ubiquitous Energy Network Connecting
Everything, EI2 2019, 631-636. https://doi.org/10.1109/EI247390.2019.9061928

Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports, 7, 8176-8186.
https://doi.org/10.1016/j.egyr.2021.08.126

Li, Y., Wei, X., Li, Y., Dong, Z., Shahidehpour, M. (2022). Detection of False Data Injection Attacks in Smart Grid: A Secure Federated Deep Learning Approach. IEEE Transactions on Smart Grid, 13(6), 4862-4872.
https://doi.org/10.1109/TSG.2022.3204796

Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?. IEEE Signal Processing Magazine, 35(5), 41-49.

Liu, J., Xiao, Y., Li, S., Liang, W., & Chen, C. L. P. (2012). Cyber security and privacy issues in smart grids. IEEE Communications Surveys and Tutorials, 14(4), 981-997.
https://doi.org/10.1109/SURV.2011.122111.00145

Long Cheng, Xuewu Chen, Jonas De Vos, Xinjun Lai, F. W. (2019). Applying a random forest method approach to model travel mode choice behavior. Travel Behaviour and Society, 14, 1-10.
https://doi.org/https://doi.org/10.1016/j.tbs.2018.09.002

Mellor, A., Boukir, S., Haywood, A., & Jones, S. (2015). Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin. ISPRS Journal of Photogrammetry and Remote Sensing,105, 155-168.
https://doi.org/10.1016/j.isprsjprs.2015.03.014

Merabet, A., Ahmed, K. T., Ibrahim, H., Beguenane, R., & Ghias, A. M. Y. M. (2017). Laboratory Scale Microgrid Based Wind-PV-Battery. IEEE Transactions on Sustainable Energy, 8(1), 145-154.

Metke, A. R., & Ekl, R. L. (2010). Security technology for smart grid networks. IEEE Transactions on Smart Grid, 1(1), 99-107. https://doi.org/10.1109/TSG.2010.2046347

Mingkui Wei, W. W. (2016). Data-centric threats and their impacts to real-time communications in smart grid. Computer Networks, 104, 174-188. https://doi.org/10.1016/j.comnet.2016.05.003

M. Elmusrati. (2022). Lecture Notes on Machine Learning Algorithms", ICAT3120 Machine Learning Course. School of Technology and Innovations, University of Vaasa.

Hasan, M. K., Habib, A. A., Shukur, Z., Ibrahim, F., Islam, S., & Razzaque, M. A. (2023). Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations. Journal of Network and Computer Applications, 209, 103540.

More, S., Hajari, S., Majeed, M.A., Singh, N.K., Mahajan, V. (2022). Cyber Security for Smart Grid: Vulnerabilities, Attacks, and Solution. Sustainable Technology and Advanced Computing in Electrical Engineering. Lecture Notes in Electrical Engineering, 939. https://doi.org/https://doi.org/10.1007/978-981-19-4364-5-60

Moslehi, K., & Kumar, R. (2010). A reliability perspective of the smart grid. IEEE Transactions on Smart Grid, 1(1), 57-64. https://doi.org/10.1109/TSG.2010.2046346

Mutani, G., Todeschi, V., Tartaglia, A., & Nuvoli, G. (2019). Energy communities in piedmont region (IT). the case study in pinerolo territory. INTELEC, International Telecommunications Energy Conference (Proceedings), 2018-Octob. https://doi.org/10.1109/INTLEC.2018.8612427

Mylrea, M., & Gourisetti, S. N. G. (2017). Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security. Proceedings - 2017 Resilience Week, RWS 2017, 18-23. https://doi.org/10.1109/RWEEK.2017.8088642

Nguyen, T. T., & Reddi, V. J. (2021). Deep Reinforcement Learning for Cyber Security. IEEE Transactions on Neural Networks and Learning Systems, 1-17. https://doi.org/10.1109/TNNLS.2021.3121870

Niculescu, S. P. (2003). Artificial neural networks and genetic algorithms in QSAR. Journal of Molecular Structure: THEOCHEM, 622(1-2), 71-83. https://doi.org/10.1016/S0166-1280(02)00619-X

Novosel, D. (2012). Experiences with deployment of smart grid projects. 2012 IEEE PES Innovative Smart Grid Technologies, ISGT 2012. https://doi.org/10.1109/ISGT.2012.6175600

Oyewole, P. A., & Jayaweera, D. (2020). Power System Security with Cyber-Physical Power System Operation. IEEE Access, 8, 179970-179982. https://doi.org/10.1109/ACCESS.2020.3028222

Pandey, M., & Kumar Sharma, V. (2013). A Decision Tree Algorithm Pertaining to the Student Performance Analysis and Prediction. International Journal of Computer Applications, 61(13), 1-5. https://doi.org/10.5120/9985-4822

Phom, H. S., Kuntze, N., Rudolph, C., Cupelli, M., Liu, J., & Monti, A. (2010). A user-centric privacy manager for future energy systems. 2010 International Conference on Power System Technology: Technological Innovations Making Power Grid Smarter, POWERCON2010, 1-7. https://doi.org/10.1109/POWERCON.2010.5666447

Prabhakar, P., Arora, S., Khosla, A., Beniwal, R. K., Arthur, M. N., Arias-Gonzáles, J. L., & Areche, F. O. (2022). Cyber Security of Smart Metering Infrastructure Using Median Absolute Deviation Methodology. Security and Communication Networks, 2022. https://doi.org/10.1155/2022/6200121

Procopiou, A., & Komninos, N. (2015). Current and future threats framework in smart grid domain. 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015, 1852-1857. https://doi.org/10.1109/CYBER.2015.7288228

Qiumei, Z., Dan, T., & Fenghua, W. (2019). Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function.IEEE Access, 7, 151359-151367. https://doi.org/10.1109/ACCESS.2019.2948112

Radoglou-Grammatikis, P. I., & Sarigiannidis, P. G. (2019). Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems. IEEE Access, 7, 46595-46620. https://doi.org/10.1109/ACCESS.2019.2909807

Rajput, G., Raut, G., Chandra, M., & Vishvakarma, S. K. (2021). VLSI implementation of transcendental function hyperbolic tangent for deep neural network accelerators. Microprocessors and Microsystems, 84(April 2020), 104270. https://doi.org/10.1016/j.micpro.2021.104270

Rohmeyer, P., & Ben-zvi, T. (2015). Managing Cloud Computing Risks in Finan-

cial Services Institutions. 2015 Portland International Conference on Management of Engineering and Technology (PICMET),
519-526. https://doi.org/10.1109/PICMET.2015.7273004

Sadiq, M., Ali, S. W., Terriche, Y., Mutarraf, M. U., Hassan, M. A., Hamid, K., Ali, Z., Sze, J. Y., Su, C. L., & Guerrero, J. M. (2021). Future Greener Seaports: A Review of New Infrastructure, Challenges, and Energy Efficiency Measures. IEEE Access, 9, 75568-75587.
https://doi.org/10.1109/ACCESS.2021.3081430

Safavian, S. R., & Landgrebe, D. (1991). A Survey of Decision Tree Classifier Methodology. IEEE Transactions on Systems, Man and Cybernetics, 21(3), 660-674. https://doi.org/10.1109/21.97458

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., & Tarantola, S. (2010). Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Computer Physics Communications, 181(2), 259-270. https://doi.org/10.1016/j.cpc.2009.09.018

Sarker, I. H. (2022a). AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems. SN Computer Science, 3(2), 1-20. https://doi.org/10.1007/s42979-022-01043-x

Sarker, I. H. (2022b). Machine Learning for Intelligent Data Analysis and Automation in Cybersecurity: Current and Future Prospects. *Annals of Data Science*. https://doi.org/10.1007/s40745-022-00444-2

Sarker, I. H., Furhad, M. H., & Nowrozy, R. (2021). AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. SN Computer Science, 2(3), 1-18.
https://doi.org/10.1007/s42979-021-00557-0

Shampa Banik, Sohag Kumar Saha, Trapa Banik, S. M. M. H. (2023). Anomaly Detection Techniques in Smart Grid Systems: A Review. 2023 IEEE World AI IoT Congress (AIIoT).

Shin, M. J., Guillaume, J. H. A., Croke, B. F. W., & Jakeman, A. J. (2013). Addressing ten questions about conceptual rainfall-runoff models with global sensitivity analyses in R. Journal of Hydrology, 503, 135-152.
https://doi.org/10.1016/j.jhydrol.2013.08.047

Siraj, V. F. and A. (2014). Applications of machine learning in cyber security. 27th International Conference on Computer Applications.

Smith, M. D., & Pate-Cornell, M. E. (2018). Cyber risk analysis for a smart grid: How smart is smart enough? A multiarmed bandit approach to cyber security investment. IEEE Transactions on Engineering Management, 65(3), 434-447. https://doi.org/10.1109/TEM.2018.2798408

Spataru, C. (2013). The future whole energy system stability, reliability and security: WITH or WITHOUT fear of blackouts? IET Seminar Digest, 2013(15377). https://doi.org/10.1049/ic.2013.0155

Steimer, P. K. (2009). Power electronics, a key technology for future more electrical energy systems. 2009 IEEE Energy Conversion Congress and Exposition, ECCE 2009, 1161-1165.
https://doi.org/10.1109/ECCE.2009.5316175

Suicimezov, N., & Georgescu, M. R. (2014). Emerging Markets Queries in Finance and Business IT Governance in Cloud. Procedia Economics and Finance, 15(14), 830-835. https://doi.org/10.1016/s2212-5671(14)00531-0

Sun, C. C., Hahn, A., & Liu, C. C. (2018). Cyber security of a power grid: State-of-the-art. International Journal of Electrical Power and Energy Systems, 99(November 2017), 45-56.
https://doi.org/10.1016/j.ijepes.2017.12.020

Svendsen, H. G., Shetaya, A. A., & Loudiyi, K. (2017). Integration of renewable energy and the benefit of storage from a grid and market perspective - Results from Morocco and Egypt case studies. Proceedings of 2016 International Renewable and Sustainable Energy Conference, IRSEC 2016, 1164-1168.
https://doi.org/10.1109/IRSEC.2016.7984007

Syrmakesis, A. D., Alcaraz, C., & Hatziargyriou, N. D. (2022). Classifying resilience approaches for protecting smart grids against cyber threats. International Journal of Information Security, 21(5), 1189-1210. https://doi.org/10.1007/s10207-022-00594-7

Taghavinejad, S. M., Taghavinejad, M., Shahmiri, L., Zavvar, M., & Zavvar, M. H. (2020). Intrusion Detection in IoT-Based Smart Grid Using Hybrid Decision Tree. 2020 6th International Conference on Web Research, ICWR 2020, 152-156. https://doi.org/10.1109/ICWR49608.2020.9122320

Taji, B., Chan, A. D. C., & Shirmohammadi, S. (2018). False Alarm Reduction in Atrial Fibrillation Detection Using Deep Belief Networks. IEEE Transactions on Instrumentation and Measurement, 67(5), 1124-1131.
https://doi.org/10.1109/TIM.2017.2769198

Tan, S., De, D., Song, W. Z., Yang, J., & Das, S. K. (2017). Survey of Security Advances in Smart Grid: A Data Driven Approach. IEEE Communications Surveys and Tutorials, 19(1), 397?422.

Tang, D., Fang, Y. P., & Zio, E. (2023). Vulnerability analysis of demand-response with renewable energy integration in smart grids to cyber attacks and online detection methods. Reliability Engineering and System Safety, 235(January), 109212. https://doi.org/10.1016/j.ress.2023.109212

Teixeira, M. A., Salman, T., Zolanvari, M., Jain, R., Meskin, N., & Samaka, M. (2018). SCADA system testbed for cybersecurity research using machine learning approach. Future Internet, 10(8). https://doi.org/10.3390/fi10080076

Tony Flick, J. M. (2010). Securing the Smart Grid Next Generation Power Grid Security. Elsevier.

Tuballa, M. L., & Abundo, M. L. (2016). A review of the development of Smart Grid technologies. Renewable and Sustainable Energy Reviews, 59, 710-725. https://doi.org/10.1016/j.rser.2016.01.011

Tufail, S., Parvez, I., Batool, S., & Sarwat, A. (2021). A survey on cybersecurity challenges, detection, and mitigation techniques for the smart grid. Energies, 14(18), 1-22. https://doi.org/10.3390/en14185894

Vaos, J., Kuhn, R., Laplante, P., & Applebaum, S. (2018). Internet of Things (IoT) Trust Concerns Draft NISTIR 822. Draft NISTIR 8222, 50-50. https://csrc.nist.gov/publications/detail/white-paper/2018/10/17/iot-trust-concerns/draft

Voas, J. (2016). Networks of "Things." NIST Special Publication, 800(183), 800-183.

Wang, W., & Lu, Z. (2013). Cyber security in the Smart Grid: Survey and challenges. Computer Networks, 57(5), 1344-1371. https://doi.org/10.1016/j.comnet.2012.12.017

Waqar, A., Hu, J., Awais, M., Xia, S., & Ai, X. (2021). Distributed Operation of Multi-Microgrids under Censored Communication. 2021 4th International Conference on Energy, Electrical and Power Engineering, CEEPE 2021, 800-806. https://doi.org/10.1109/CEEPE51765.2021.9475633

Workman, M., Bommer, W. H., & Straub, D. (2008). Security lapses and the omission of information security measures: A threat control model and empirical test. Computers in Human Behavior, 24(6), 2799-2816. https://doi.org/10.1016/j.chb.2008.04.005

Wulf, F., Strahringer, S., & Westner, M. (2019). Information security risks, benefits, and mitigation measures in cloud sourcing. Proceedings - 21st IEEE Conference on Business Informatics, CBI 2019, 1, 258-267. https://doi.org/10.1109/CBI.2019.00036

Liu, X., & Nielsen, P. S. (2016). Regression-based online anomaly detection for smart grid data. arXiv preprint arXiv:1606.05781.

Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A survey on cyber security for smart grid communications.IEEE Communications Surveys and Tutorials, 14(4), 998-1010.
https://doi.org/10.1109/SURV.2012.010912.00035

Yang, Y., Littler, T., Sezer, S., McLaughlin, K., & Wang, H. F. (2011). Impact of cyber-security issues on Smart Grid. IEEE PES Innovative Smart Grid Technologies Conference Europe, 1-7.
https://doi.org/10.1109/ISGTEurope.2011.6162722

Ye Yan, Yi Qian, Hamid Sharif, and D. T. (2013). A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 15(1). https://doi.org/10.1109/SURV.2012.021312.00034

Yeboah-Ofori, A., & Islam, S. (2019). Cyber security threat modeling for supply chain organizational environments. Future Internet, 11(3). https://doi.org/10.3390/fi11030063

Yigit, M., Gungor, V. C., & Baktir, S. (2014). Cloud Computing for Smart Grid applications. Computer Networks, 70, 312-329. https://doi.org/10.1016/j.comnet.2014.06.007

Youssef, T. A., Hariri, M. El, Elsayed, A. T., & Mohammed, O. A. (2018). A DDS-Based Energy Management Framework for Small Microgrid Operation and Control. IEEE Transactions on Industrial Informatics, 14(3), 958-968.
https://doi.org/10.1109/TII.2017.2756619

Z Su, L Xu, S Xin, W Li, Z Shi, Q. G. (2017). A future outlook for cyber-physical power system. 2017 IEEE Conference on Energy Internet and Energy System.

Zhang, Q., Sun, Y., & Cui, Z. (2010). Application and analysis of ZigBee technology for Smart Grid. Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application, 171-174. https://doi.org/10.1109/ICCIA.2010.6141563

Zhang, Y., Huang, T., & Bompard, E. F. (2018). Big data analytics in smart grids: a review. Energy Informatics, 1(1), 1-24. https://doi.org/10.1186/s42162-018-0007-5

Publication I

Soft Computing (2022) 26:13109–13118
https://doi.org/10.1007/s00500-022-06761-1

FOCUS

On the performance metrics for cyber-physical attack detection in smart grid

Sayawu Yakubu Diaba[1] · Miadreza Shafie-khah[2] · Mohammed Elmusrati[1]

Accepted: 3 January 2022 / Published online: 21 January 2022

Abstract
Supervisory Control and Data Acquisition (SCADA) systems play an important role in Smart Grid. Though the rapid evolution provides numerous advantages it is one of the most desired targets for malicious attackers. So far security measures deployed for SCADA systems detect cyber-attacks, however, the performance metrics are not up to the mark. In this paper, we have deployed an intrusion detection system to detect cyber-physical attacks in the SCADA system concatenating the Convolutional Neural Network and Gated Recurrent Unit as a collective approach. Extensive experiments are conducted using a benchmark dataset to validate the performance of the proposed intrusion detection model in a smart metering environment. Parameters such as accuracy, precision, and false-positive rate are compared with existing deep learning models. The proposed concatenated approach attains 98.84% detection accuracy which is much better than existing techniques.

Keywords Supervisory control and data acquisition (SCADA) systems · Intrusion detection system (IDS) · Industrial control system (ICS) · Cyber-physical security · Smart grid · Convolutional neural network (CNN) · Gated recurrent unit (GRU)

1 Introduction

Most of the Intrusion Detection Systems (IDS) used in Supervisory Control and Data Acquisition (SCADA) in power distribution networks are currently concentrated on the cyber sector by ignoring the process states in the physical field (Rakas et al. 2020). Attacks on protocol traffic are being detected, but attacks on processes like Replay and Man-in-the-Middle (MITM) attacks are complex to detect. The performance criteria, risk management requirements, and coordination requirements vary between Information Technology System (IT System) and Industrial Control System (ICS) networks. In an IT system, a long delay could be appropriate, but coming to ICS, reaction time is crucial (Ghosh and Sampalli 2019). In IT systems, data security and integrity are most significant, whereas fault tolerance is less significant, while in ICS, human safety is most crucial, followed by process security, and fault tolerance is necessary (Paridari et al. 2018). Many specific communication protocols without ID certification, encryption, and timestamps were utilized in ICS, whereas standard communication protocols are utilized in IT systems. Zero-day, Denial of Service (DoS), Replay and MITM attacks to ICS will trigger the above-mentioned delay, fault, and information leakage caused by protocols (Hu et al. 2019).

Cyber-physical systems are widely used to integrate the physical process and computations so that the system can be controlled effectively. The performance of the system relies on proper control of its elements, like sensors and actuators. Efficient and secure communication between the system element is most important as it directly affects the

Communicated by Joy Iong-Zong Chen.

✉ Sayawu Yakubu Diaba
saywu.diaba@student.uwasa.fi

Miadreza Shafie-khah
mshafiek@uwasa.fi

Mohammed Elmusrati
moel@uwasa.fi

[1] School of Technology and Innovation, Department of TelecommunicationsEngineering, University of Vaasa, Vaasa, Finland

[2] School of Technology and Innovations, Department of Electrical Engineering, University of Vaasa, Vaasa, Finland

system performance. Malfunctioning the device characteristics might lead to a serious issue in industrial control systems. The system elements face serious security threats which affect the sensing and data actuation. Attacks on IT system networks cause congestion or data leakage, but attacks on ICS networks may result in both data leakage as well as harm to physical infrastructure. As a result, for the SCADA systems, which are commonly utilized in the power distribution networks to ensure the security of the controlled processes, cyber-security is considered a significant part of SCADA (Zhang et al. July 2019). The protection of communication protocols, asset management, physical infrastructures, and controlled processes will come under the security of the SCADA system that is the most important element of the smart grid, and these cannot be handled the same as IT system contemporaries. Some of the key components are supporting software such as Human Machine Interface (HMI), Distributed Control Systems (DCS), Programmable Logical Controllers (PLC), Remote Terminal Units (RTU), network equipment, servers, and computers (Cómbita et al. 2019, Pang et al. 2020, Sun et al. 2020, Elnour et al. 2020). Hence, it is essential to protect the system against attacks and secure communication.

Intrusion detection systems are used to detect the security threats and attacks in a system where the systems can able to detect but not able to prevent the attacks. However, by training the detection systems properly the attacks can be detected efficiently without any manual intervention which may reduce the huge loss compared to the loss acquired in a system without an intrusion detection system. These systems will work as a second-line defense in any architecture and plays a vital role in cyber-physical systems to detect different types of attacks. Intrusion detection systems classify normal and abnormal behavior which helps the system to detect unknown attacks. This essential feature is adopted for cyber-physical systems. A wide range of devices and dynamic computing resources, different software, and operating systems are generally included in cyber-physical systems. Detecting intrusion in such systems using machine learning algorithms-based models is crucial due to the heterogeneous deployment nature. Obtaining labels for attacks can be very time-consuming, challenging, and sometimes even impossible. Therefore, unsupervised learning techniques, capable of detecting cyber-attacks without a need for labels, are deemed best for this task (Keshk et al. 2021, Gumaei et al. 2020). However, the most existing unsupervised techniques are not able to deal with the nonlinearity and inherent correlations of multivariate time series, which represent a considerable amount of real-world data, including data streams generated by sensors in CPSs (Hu et al. 2019; Rodofile et al. 2019; Homay et al. 2020). Therefore, a new unsupervised technique independent from any prior knowledge of cyberattacks is needed to detect intrusions in CPSs.

The major contributions of this paper are summarized as follows.

(1) CNN and GRU are combined to obtain an intrusion detection system for detecting attacks in smart grid metering infrastructure.
(2) An intense experimental analysis is presented using benchmark datasets to obtain improved accuracy and detection rate performance for the proposed model.

The rest of the paper is arranged as follows: a brief literature analysis is presented in Sect. 2; the proposed intrusion detection model is presented in Sect. 3. Experimental results and observations are discussed in Sect. 4 and finally, the conclusion is presented in Sect. 5.

2 Related works

Recent research works in industrial systems and their evolutions are discussed in this section. Intrusion detection is the major objective and the research directed toward analyzing the features of existing intrusion detection systems. The authors of Khan et al. (2019) have introduced a new method called anomaly detection for ICS. This method utilized a hybrid approach by taking the benefits of the reliable and predictable nature related to communication patterns, which perform in-ground devices in ICS. Initially, few preprocessing approaches were implemented for scaling and standardizing the data. To enhance the performance of anomaly detection, dimensionality reduction algorithms were used. Later, the nearest-neighbor rule algorithm was utilized for balancing the dataset. A signature database was created by noting the system in a time using a bloom filter. Subsequently, a hybrid approach was created for anomaly detection by combining the instance-based learner and package contents-level detection. Here, the developed model has attained the best results when compared to other state-of-the-art models.

The authors of Qian et al. (2020) have suggested a method in a physical way as well as a cyber-way for attack detection. In order to detect malicious behaviors for physical component prevention, process states validation was utilized and being damaged by Zero-day, MITM, and Replay attacks. For branching shaped data sets classification, a nonparallel hyperplane-based fuzzy classifier was developed that was quite complex, complex to classify using two parallel hyperplanes of the Support Vector Machine (SVM) to detect DoS and other cyber-attacks. To test the developed model and validation part, Modbus/Transmission Control Protocol (TCP) traffic data and

simulation process states were used. Thus, it has been proved that the suggested approach was superior to other approaches.

In (Sheng et al. 2021), a cyber-physical technique in the SCADA system for intrusion detection has analyzed the risk levels faced in industries. This was utilized to characterize the structure of the network and SCADA system's process by the extraction and correlation of communication patterns and the ICS device condition. If any violation occurs, then this was considered as an abnormal behavior that was caused due to network attacks. A risk estimation approach was suggested to measure the damage degree of the attack on the infrastructure by associating network intrusions with the state of the SCADA system, providing network teams with more knowledge regarding network attacks. Furthermore, the proposed approach outperformed existing approaches in identifying and evaluating numerous cyber-attacks against the SCADA system.

Privacy-Preserving Anomaly Detection framework named PPAD-CPS is reported in the research work of Keshk et al. (Keshk et al. 2021). The aim is to secure confidential data and discover malicious attacks in power systems and their network traffic. This framework included two modules namely data preprocessing and anomaly detection modules. To filter and transform the real data into a new kind of data, a data preprocessing module was recommended that has attained privacy preservation target. By using Kalman Filter (KF) and Gaussian Mixture Model (GMM), an anomaly detection model was employed for analyzing the posterior probabilities of anomalous and legitimate events. Two public datasets such as UNSW-NB15 and Power System were used for analyzing the proposed framework. This analysis has been proved that the developed PPAD-CPS outperformed the existing methodologies.

In (Kalech 2019) cyber-attack detection models based on temporal pattern recognition (TPR), which searched for anomalies in the data sent by the SCADA elements in the network and found anomalies that were occurred when legal commands like incorrect and unauthorized time intervals were misused. Artificial Neural Networks (ANN) and Hidden Markov Model (HMM) were suggested for evaluating the performance. The evaluation was done on both simulated and real SCADA data using five various feature extraction approaches. The outcomes have shown that TPR models were performed well in detecting cyber-attacks.

Gumaei et al. (Gumaei et al. 2020) have proffered a new security control method for cyber-attack detection in smart grid, which merged feature detection and reduction models for decreasing the features count and attained a better detection rate. For eliminating irrelevant features, the Correlation-based Feature Selection (CFS) technique was utilized that enhanced the detection rate. With the help of optimal features that were selected, cyber and normal attack events were classified using the Instance-Based Learning (IBL) algorithm. By using public datasets of SCADA power system, the experiments were performed relied on tenfold cross-validation approach. This has been revealed that the suggested model consisted of huge detection rate.

Rodofile et al. (Rodofile et al. 2019) have presented a cyber-attack structure, which detects attacks in SCADA systems. The developed model recognized "traditional IT-based attacks, protocol specific attacks, configuration-based attacks and control process attacks" for describing the practical attacks. The recognition of attacks in the whole system has an advantage of allowing us to protect over them with more effectiveness and awareness. A case study was presented by illustrating the sequence of attacks on Distributed Network Protocol 3 (DNP3), which facilitated to affirm the reliability of the developed model.

Reference (Homay et al. 2020) has implemented a robust security control solution as a logic level security on DCS and SCADA systems. In order to establish trust among DCS device components, the developed model ensured message integrity, but this was not considered as the protection layer on industrial automation systems. Malicious attacks like Stuxnet were avoided by the developed solution called low-level security process. From the analysis, the following points are observed as research gaps. The security of SCADA systems has been disrupted using earlier IC system attacks. The significance of defending and securing ICS networks has increased attacks on critical infrastructures. The features and challenges of various methodologies of detecting and blocking cyber-security attack on SCADA systems that has existed earlier are given subsequently. HML-IDS can detect anomalies very fast and it can detect unseen attacks also it can deal with data samples that seem to be hybrid which is complex. But it has some demerits like enhancement of detection rate (Feng et al. 2017).

NHFC is capable of modeling a given problem into any degree of accuracy and has high detection accuracy in detecting zero-day, Replay, and MITM attacks. It needs to improve in detecting the attacks for securing the manufacturing process (Wang et al. August 2020). Cyber-physical model is used to detect the network intrusions. It has high accuracy. Though, it is not considered as the secure appliance, because of the lack of multi-factor authentication models. PPAD-CPS has huge privacy levels. It attains the best accuracy, detection rate, and processing times. It needs to perform a principal and independent component analysis for transforming the high-dimensional space into low-dimensional space in order to enhance the performance. TPR can detect cyber-attacks including

legitimate functions. It has high detection accuracy. It needs to reduce the count of false alarms by considering PLC identity (Dhaya 2021; Jacob and Ebby Darney 2021; Haoxiang and Smys 2021; Smys et al. 2021).

CFS-KNN resorts to various correlation measures for removing the irrelevant features and retaining those using predictive power. It is robust to exploit inter-feature relationships. It is sensitive to noise and has an over-fitting problem, which leads to reduce the system performance. DNP3 consists of efficient Internet Provider Security (IPS) and IDS technologies. It can combine four categories for describing the practical attacks. However, troubleshooting the system is quite complex because of distribution over many servers.

To overcome these limitations concatenated deep learning approach is presented in this paper. Deep learning approach has the ability to process high-dimensional data and produce better results than machine learning approaches. (Fig. 1)

3 Proposed work

The proposed concatenated deep learning model is presented in this section to detect intrusions in Cyber-Physical Systems. The simple environment considered for this paper is depicted in Fig. 2. The framework consists of two entities such as industrial agents and cloud servers. The industrial agents are the industrial Cyber-Physical System owners and they oversee the local intrusion detection model. Collecting industrial cyber-physical system data and updating the essential parameters for intrusion detection are some of the regular activities of industrial agents. Industrial agents interact with cloud servers to update the data. Whereas cloud servers take the responsibility of building a comprehensive intrusion detection system using the model parameters obtained from the locally trained model in the industrial agents. The final intrusion detection model could be obtained through multiple interactions between each agent and cloud server.

The threat model we have considered for analysis includes four different threats such as reconnaissance attacks, command injection attacks, response injection attacks, DoS attacks. Cyber threats targeting industrial cyber-physical security are considered for the proposed concatenated deep learning-based intrusion detection system. The reconnaissance attacks are performed to collect industrial cyber-physical security system valuable information. Network architecture details, device features, network protocols are the major aim of the intruder to perform reconnaissance attacks. To mislead or deviate the industrial physical security system behavior, command injection attacks are performed. In this attack, the intruder injects some false information to control a system or provides wrong configuration commands to collapse the system behavior. Unauthorized access, invalid communications, wrong set points are the outcomes of a command injection attack.

To monitor and observe the remote process state in the industrial cyber-physical security system, response injection attacks are performed. These attacks interfere with the system process and provide false responses to the service queries which affect the system state. DoS attacks are quite common and familiar in the network, in the case of industrial cyber-physical security systems the attacker flooded the targets with redundant requests to deplete the server resources in the industrial cyber-physical security system. Due to these boundless requests, the system prevents legitimate requests which affect the system services. Figure 3 depicts the proposed concatenated deep learning

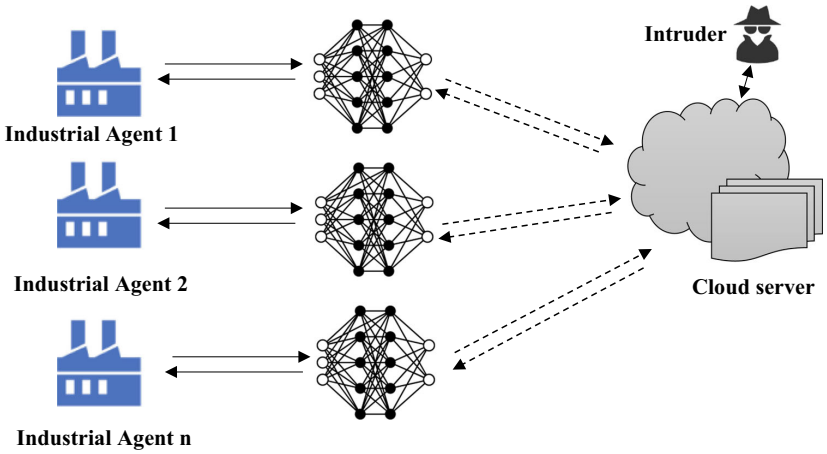


Fig. 1 Proposed system model

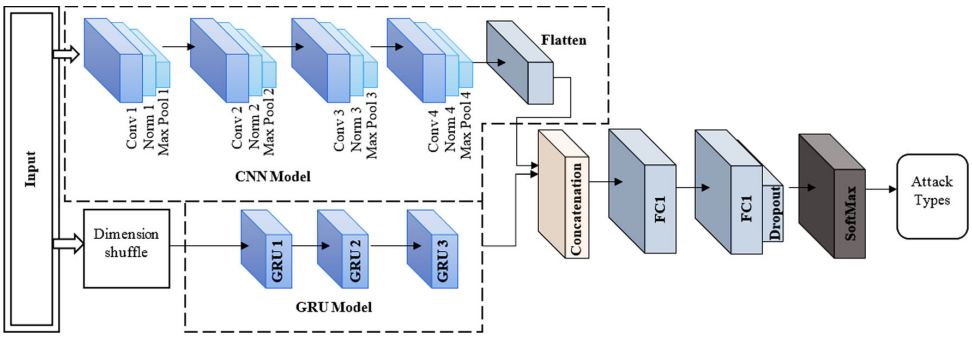


Fig. 2 Proposed concatenated deep learning model

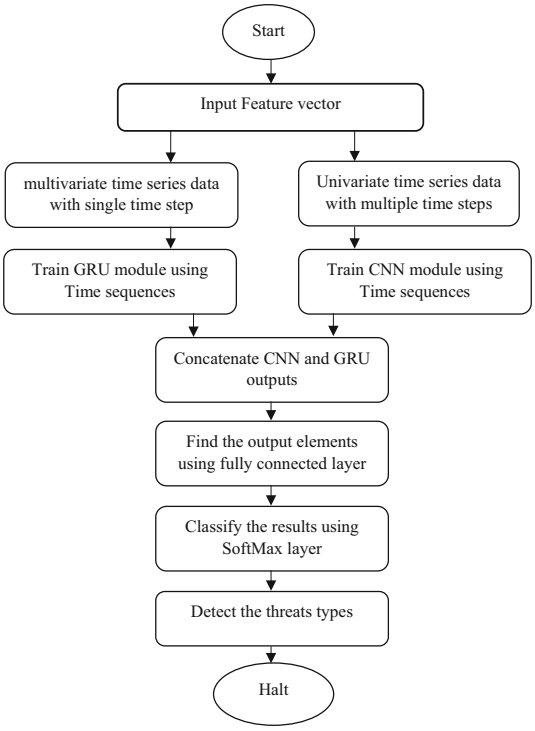


Fig. 3 Process flow of proposed Intrusion detection model

model for intrusion detection in the industrial cyber-physical security system.

The proposed intrusion detection model includes a CNN model and a GRU unit module. The outputs of both modules are combined and processed by a fully connected layer followed by the SoftMax layer. The building block of the convolutional network model includes four convolution blocks. Each block includes a convolutional layer, a max-pooling layer. Batch normalization is performed between the convolution layer and max-pooling layer. The GRU is comprised of three GRU layers. The results of the CNN and GRU model are concatenated and then processed using two fully connected layers. In order to prevent over-fitting, a dropout layer is included after the fully connected layer. Finally, the SoftMax layer is used to map the output of fully connected to probability distribution and predicts the attack types.

3.1 System model

Consider a feature vector v as input for the proposed model which is a one-dimensional vector function representing the numerical features of industrial data. The input is processed by CNN and GRU model. GRU is a modified LSTM model which includes a gated recurrent neural network. LSTM consists of three gates such as forget gate, input gate, and output gate whereas GRU comprises of two gates such as update gate and reset gate. Due to this it requires less parameter for training which provides quick convergence compared to LSTM. Owing to this reason instead of LSTM, GRU is adopted in the proposed design. The long dependency features are captured by the GRU module and learn essential information from the historical data using memory cell. The reset gate is used to forget or remove unnecessary information. Generally, the input for GRU module will be time sequence data and the given input feature vector is a multivariate time series data with a single time step. Therefore, prior to GRU module, a dimension shuffling process is performed that transposes the dimensions of the feature vector v. The dimension shuffling is given as

$$\tilde{v} = \mathrm{shuf}(v) \tag{1}$$

The GRU module process the dimensional shuffled data $\tilde{v}$ and produces output. The output of first layer is given as input to the second layer and it repeats. The output of GRU module is obtained using two activation functions such as

tan$h$ and σ. The essential formulations observed for the GRU module is summarized as follows

$$\Gamma_U = \sigma\left(\omega_U\left[\tilde{v}^{(t-1)}, x^{(t)}\right] + b_U\right) \tag{2}$$

$$\Gamma_R = \sigma\left(\omega_R\left[\tilde{v}^{(t-1)}, x^{(t)}\right] + b_R\right) \tag{3}$$

where Γ_U and Γ_R represents the update gate and reset gate respectively. The range of $\Gamma_U \in \{0, 1\}$ and the range of $\Gamma_R \in \{-1, 1\}$. ω_U and ω_R are the weight functions of update and reset gate. b_U and b_R represents the bias vectors for update and reset gate correspondingly. The candidate activation function for the recurrent unit is given as

$$\tilde{v}^{(t)} = \tanh\left(\omega_v\left[\Gamma_R \times \tilde{v}^{(t-1)}, x^{(t)}\right] + b_v\right) \tag{4}$$

where ω_v are the weight functions of activation function, b_v is the bias vector and $x^{(t)}$ is the inputs of training data. The output of a single GRU unit is given as

$$v^{(t)} = \left((1 - \Gamma_U) \times \tilde{v}^{(t-1)}\right) + \left(\Gamma_U \times \tilde{v}^{(t)}\right) \tag{5}$$

where $\tilde{v}^{(t-1)}$ is the current unit input which is obtained from the previous unit output. The final output of GRU module is given as I. The same input used for GRU model is parallelly provided to CNN module. The input feature vector v is considered as a univariate time series data with multiple time steps. Since CNN is suitable to process high-dimensional data it does not require any dimensional shuffling module as like GRU. One-dimensional layer is used in the proposed model and the convolution operation is represented as

$$h_1 = \text{convblock1}(v) \tag{6}$$

$$h_2 = \text{convblock2}(h_1) \tag{7}$$

$$h_3 = \text{convblock3}(h_2) \tag{8}$$

$$h_4 = \text{convblock4}(h_3) \tag{9}$$

where h_1, h_2, h_3 and h_4 are the hidden vectors. After each convolution layer, a batch normaliztion and pooling layer is included. The normalization layer normalizes the features which is obtained after the convolution proces and the pooling layer is used to reduce the dimensionality of the data. There are two types of pooling functions such as max pooling an average pooling. In this research work, average pooling is used in the pooling layer. The reduced output from the pooling layer is provided as the input to the next convolution block. The average pooling layer is mathmatically expressed as

$$x^i = \text{avg}P\left(x^{i-1}\right) \tag{10}$$

where x^i is the ouput of pooling layers, and x^{i-1} is the previous values obtained from the convolution layer. i represents the number of pooling layers. The final output of convolution layer is given to flatten layer that converts the data into one-dimensional vector and it is given as

$$J = \text{flatten}(h_4) \tag{11}$$

The output I from the GRU module and J from the CNN module are concatenated which is described as $c_t = \text{concate}(I, J)$. Followed by two fully connected layers are used in the proposed design. To prevent data overfitting a dropout function is used. Then the dropout layer, the final SoftMax layer provides the classification results which provide the attack types. The SoftMax function is described as

$$\hat{y} = \text{softmax}(\varphi) \tag{12}$$

where φ is the output of dropout layer. In order to evaluate the loss function for the proposed model cross-entropy function is used and it is given as

$$l = -\frac{1}{b}\sum_{i=1}^{n} y_i \log y_i^{'} \tag{13}$$

where the batch size is given as b, n denotes the training sample size, the predicted value is represented as $y_i^{'}$ athe nd actual value is represented as y_i. The process flow of the proposed approach is depicted in Fig. 3.

Figure 3 depicts the process flow of the proposed intrusion detection model. Initially, the process starts with the selection of input feature vectors. The multivariate time series data with a single time step is used to train the GRU model and univariate series data with multiple time steps is used to train the CNN model. The output features of each model are concatenated, and the elements are provided as input to the fully connected layer. The final classified results are obtained from the SoftMax layer and they provide the details of threats and their types.

4 Results and discussion

The proposed intrusion detection model is being experimented on a smart metering environment and the model is comprised of three network configurations, such as Home Area Network (HAN), Wide Area Network (WAN), and Neighborhood Area Network (NAN). The household applications, concentrators, smart electricity meters, data processing centers, and nodes are included in the smart infrastructure. Wireless communication or wired communication is used for communicating the elements. The communication is bidirectional and mainly internal communications are performed through HAN, which is vulnerable to attacks. A DoS attack and a probing attack are the major attacks on HAN. The NAN is used for short-

distance communication, and it is vulnerable to the user to Root (U2R) attack. WAN is used for long-distance communication and it is vulnerable to Remote to Local (R2L) attacks.

To evaluate the performance of the proposed intrusion detection system with a standard benchmark dataset, the NSL-KDD dataset is used. The data include 125,973 attacks and normal data which are provided as input to the proposed intrusion detection. Deep learning models are not able to process character-based features, so to simplify and process the input data, preprocessing steps such as normalization and feature screening are performed before the input is fed into the deep learning model. The attack data present in the dataset is categorized into four types, such as DoS, Probe attack, U2R attack, and R2L attack. The subclasses of attacks are 39 and they cover the attack types of smart metering infrastructure. The experimentation process trains and tests the data in the ratio of 80:20 according to the fivefold cross-validation method. The dataset distribution for the NSL-KDD dataset is listed in Table 1.

In the data preprocessing, the characteristic features are converted into numerical values, specifically as Eigenvectors, using a one-hot encoding process. For this process, flag features, services, and protocol types are considered in the dataset. The protocol type considers attributes such as user datagram protocol (UDP), TCP, and Internet control message protocol (ICMP). The numerical data and one-hot encoding represent the feature vectors in 1×3 dimension as like (0,0,0), (0,1,1), etc. The service feature has 70 attributes and flag features include 11 attributes and these attributes. The features after preprocessing are mapped into numerical features and combined with existing numerical features in the dataset. Also, the labels in the dataset are numerically processed such that, normal behavior is represented as '0', DoS is represented as '1', Probe label is represented as '2', R2L as '3', and U2R as '4'. In order to reduce the feature differences, the dataset is normalized and uniformly mapped. The interval range of uniform mapping is [0, 1].

Table 1 Dataset Distribution of NSL-KDD

Type	Total	Training set	Test set
Normal	67,343	53,874	13,469
Dos	45,927	36,742	9185
Probing	11,656	9325	2331
R2L	995	796	199
U2R	52	42	10
Total	1,25,973	100,778	25,195

The proposed intrusion detection model is implemented in Spyder3.0 (Python3.6) operating on Windows 10 OS installed on an i5 Intel processor at 4.20 GHz and 8 GB of memory. The learning rate was set at 0.006 and the number of epochs was 100. The hyperparameter details used in the proposed work are listed in Table 2.

The proposed model performance is further evaluated based on the convergence ability and classification performance in terms of accuracy, detection ratio, and false-positive rate. The training loss and epoch for the proposed concatenated model are depicted in Fig. 4. It is observed from the results. The training loss is gradually decreased and stable after the eighth epoch. This indicates the selection of hyperparameters is reasonable and validates the convergence ability of the proposed model. The confusion matrix for the proposed approach is depicted in Table 3.

The performance of the proposed model in terms of precision, detection rate, f1-score, and false-positive rate is depicted for the four types of attacks in Fig. 5. Based on the values obtained from the confusion matrix, the parameters are evaluated.

From Fig. 5, it can be observed that the detection abilities for DoS and Probe are above 95%, whereas the detection abilities for U2R and R2L are below 90% and 50% due to the limited number of training data. The performance of the proposed model is validated using fivefold cross-validation and the confusion matrix obtained after fivefold validation is depicted in Table 4.

The performance of the proposed model in terms of precision, detection rate, f1-score, and false-positive rate is depicted for the four types of attacks in Fig. 6. Based on the values obtained from the confusion matrix given in Table 4, the parameters are evaluated. It is observed from the results that the detection rate of the proposed model is maximum for DoS and Probe attacks and it has been reduced for R2L and U2R attacks. The reduced performance is due to the minimum number of samples in the

Table 2 Hyperparameter settings

S. no	Parameter	Filter/Neurons
1	Number of filters in CNN	8
	Number of Neurons in CNN and ReLu	16
2	Number of Hidden nodes	60
3	Activation function	ReLU
4	Dense layer	256
5	Cost function	Cross entropy
6	Batch size	128
7	Epoch	100

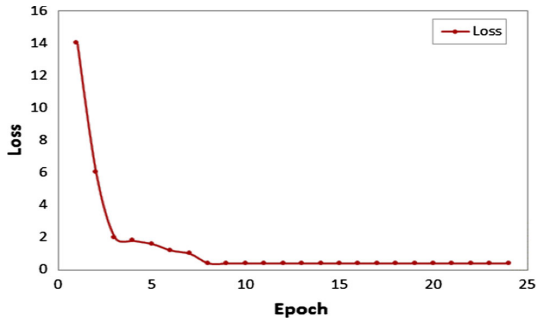


Fig. 4 Training loss vs Epochs

Table 3 Confusion matrix

True Class \ Predicted class	Normal	Dos	Probe	R2L	U2R	
Normal	13358	1	2	1	1	99.78%
Dos	2	9204	0	0	0	99.94%
Probe	5	0	2380	0	0	99.52%
R2L	11	0	0	200	0	90.11%
U2R	2	0	0	0	4	50%
	99.70%	99.73%	98.84%	99.01%	66.78%	
	Normal	DoS	Probe	R2L	U2R	

R2L and U2R attacks whereas for DoS and Probe attacks the number of samples is sufficient to obtain the desired training accuracy which improves the test accuracy.

The performance of the proposed model is compared with existing deep learning techniques like Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and Long short-term memory (LSTM) based intrusion detection models. The performance of non-concatenated models is included in the experimental analysis. The results of CNN and GRU are considered as the results of non-concatenated systems as the results are obtained separately by applying the models without any feature fusion. Results clearly depict that the non-concatenated CNN and GRU model exhibits less performance than the proposed model for all the parameters.

The parameters like precision and detection rate (Recall) are considered for analysis, and it is depicted in Figs. 7 and 8. It is observed from the results the performance of the proposed model is better compared to other models. The maximum precision and detection rate attained by the proposed model indicates that all the normal and abnormal activities in the network are detected effectively. The average precision attained by the proposed model considering all the normal and attack categories is 93.6% whereas

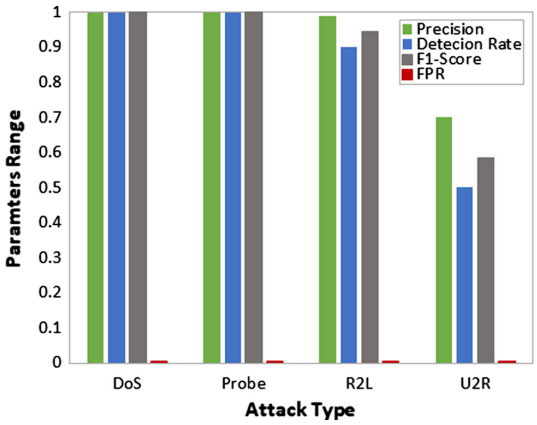


Fig. 5 Performance evaluation of proposed model

Table 4 Confusion matrix

True Class \ Predicted class	Normal	Dos	Probe	R2L	U2R	
Normal	26814	8	9	7	2	99.84%
Dos	0	18404	0	0	0	99.98%
Probe	15	0	4534	0	0	99.32%
R2L	44	0	0	310	0	78.54%
U2R	5	0	0	0	12	49%
	99.54%	99.89%	98.94%	95.84%	66.74%	
	Normal	DoS	Probe	R2L	U2R	

LSTM attains 89.1% and GRU attains 85.8% and 84% attained by the CNN-based detection model.

Based on the precision and detection rate values obtained in the previous analysis, the performance is measured in terms of F1-score and depicted in Fig. 9. The maximum F1-score attained by the proposed model indicates the maximum detection performance compared to existing deep learning methods. The overall accuracy based on the above parameters is calculated for the proposed model and existing models and depicted in Fig. 10. It is observed that the maximum accuracy is attained by the proposed model. 98.84% is the acquired detection accuracy of the proposed model whereas the existing techniques like LSTM, GRU, and CNN attain accuracy of 94.11%, 96.65%, and 97.07%, respectively. Due to efficient feature selection and concatenated process, the proposed model exhibits maximum accuracy compared to other models.

From the results, it can be observed that the proposed model efficiently detects attacks on the network and provides better detection rate and accuracy. The results were

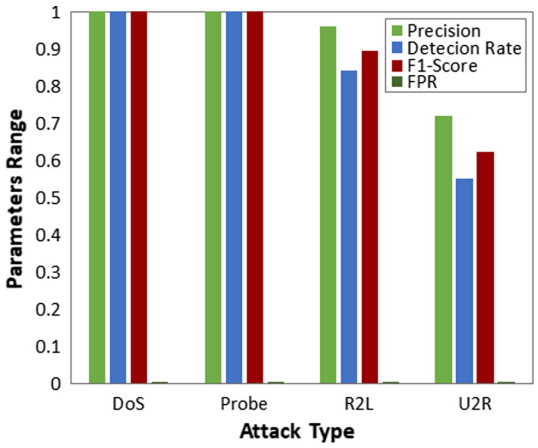


Fig. 6 Performance evaluation of proposed model

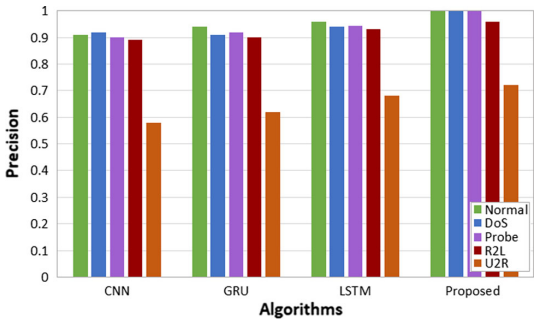


Fig. 7 Precision analysis

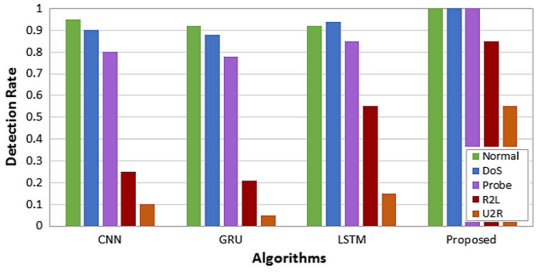


Fig. 8 Detection rate analysis

obtained for the standard dataset and the same performance can be expected in real-time smart grid data. The computational complexity of the proposed model is slightly above the mark than the existing techniques due to the initial parameters for two models used in the setup. However, for an industrial system intrusion detection model the proposed model is sufficient to detect the attacks efficiently. There may be a slight change in the detection performance due to

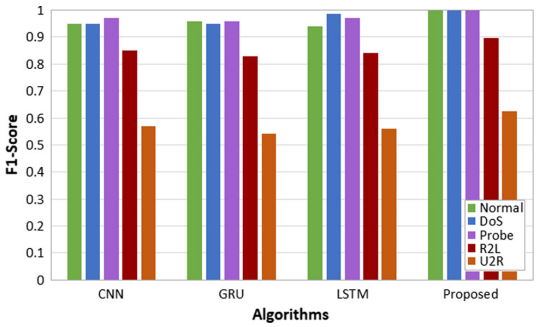


Fig. 9 F1-Score analysis

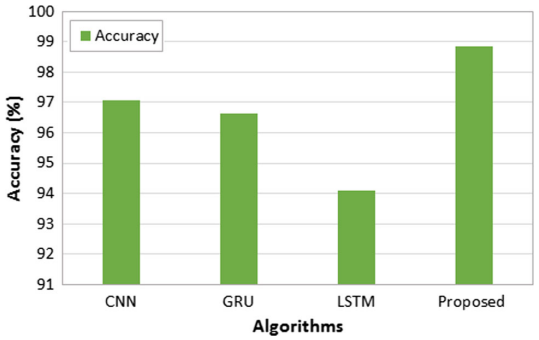


Fig. 10 Comparative analysis of Accuracy

environmental and system changes, which is the minor limitation of this paper.

5 Conclusion

This paper presents an efficient intrusion detection system for cyber-physical attack detection in the smart grid metering infrastructure. Cyber physical attacks on the SCADA systems are considered for the smart grid metering infrastructure and various types of attacks are identified. The attack types are related to standard benchmark dataset types and evaluation is performed to avoid real-time computational complexities. The NSL-KDD dataset is used for experimentation and the performance is evaluated in terms of accuracy, detection rate, precision, and false positive rate. Existing methods such as CNN, GRU, LSTM are compared with the proposed model and the results clearly demonstrate that the performance of the proposed model is superior to others. Further, this paper can be improved by focusing on the other parameters related to the grid environment.

Acknowledgements Sayawu Yakubu Diaba would like to thank the Evald and Hilda Nissi Foundation for awarding me scholarship.

Funding Open Access funding provided by University of Vaasa (UVA). No funding was received to assist with the preparation of this manuscript.

Declarations

Conflict of interests The authors declare that they have no conflict of interest to declare.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Cómbita LF, Cárdenas ÁA, Quijano N (2019) Mitigating sensor attacks against industrial control systems. IEEE Access 7:92444–92455

Dhaya R (2021) Light weight CNN based robust image watermarking scheme for security. J Inf Technol Digital World 3(2):118–132

Elnour M, Meskin N, Khan K, Jain R (2020) A Dual-isolation-forests-based attack detection framework for industrial control systems. IEEE Access 8:36639–36651

Feng C, Li T, Chana D (2017) Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks. In: 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp 261–272

Ghosh S, Sampalli S (2019) A survey of security in SCADA networks: current issues and future challenges. IEEE Access 7:135812–135831

Gumaei A, Hassan MM, Huda S, Hassan MdR, Camacho D, Ser JD, Fortino G (2020) A robust cyberattack detection approach using optimal features of SCADA power systems in smart grids. Appl Soft Comput 96:106658

Haoxiang W, Smys S (2021) A survey on digital fraud risk control management by automatic case management system. J Electr Eng Autom 3(1):1–14

Homay A, Chrysoulas C, El Boudani B, Sousa MD, Wollschlaeger M (2020) A security and authentication layer for SCADA/DCS applications. Microprocess Microsyst 6:103479

Hu Y, Sun Y, Wang Y, Wang Z (2019) An enhanced multi-stage semantic attack against industrial control systems. IEEE Access 7:156871–156882

Jacob IJ, EbbyDarney P (2021) Design of deep learning algorithm for IoT application by image based recognition. J ISMAC 3(3):276–290

Kalech M (2019) Cyber-attack detection in SCADA systems using temporal pattern recognition techniques. Comput Secur 84:225–238

Keshk M, Sitnikova E, Moustafa N, Hu J, Khalil I (2021) An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. IEEE Trans Sustain Comput 6(1):66–79

Khan IA, Pi D, Khan ZU, Hussain Y, Nawaz A (2019) HML-IDS: a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. IEEE Access 7:89507–89521

Pang Y, Xia H, Grimble MJ (2020) Resilient nonlinear control for attacked cyber-physical systems. IEEE Trans Syst, Man, Cybern: Syst 50(6):2129–2138

Paridari K, O'Mahony N, El-Din Mady A, Chabukswar R, Boubekeur M, Sandberg H (2018) A framework for attack-resilient industrial control systems: attack detection and controller reconfiguration. Proceedings of the IEEE 106(1):113–128

Qian J, Du X, Chen B, Qu B, Zeng K, Liu J (2020) Cyber-physical integrated intrusion detection scheme in SCADA system of process manufacturing industry. IEEE Access 8:147471–147481

Rakas SVB, Stojanović MD, Marković-Petrović JD (2020) A Review of research work on network-based SCADA intrusion detection systems. IEEE Access 8:93083–93108

Rodofile NR, Radke K, Foo E (2019) Extending the cyber-attack landscape for SCADA-based critical infrastructure. Int J Crit Infrastruct Prot 25:14–35

Sheng C, Yao Y, Fu Q, Yang W (2021) A cyber-physical model for SCADA system and its intrusion detection. Computer Netw 185:107677

Smys S, Vijesh Joe C (2021) Metric routing protocol for detecting untrustworthy nodes for packet transmission. J Inf Technol 3(2):67–76

Sun Q, Zhang K, Shi Y (2020) Resilient model predictive control of cyber-physical systems under DoS attacks. IEEE Trans Industr Inf 16(7):4920–4927

Wang C, Wang B, Liu H, Qu H (2020) Anomaly detection for industrial control system based on autoencoder neural network. Wireless Commun Mobile Comput 2020:3

Zhang F, Kodituwakku HADE, Hines JW, Coble J (2019) Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. IEEE Trans Industr Inf 15(7):4362–4369

Publication II

Neural Networks 159 (2023) 175–184

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Proposed algorithm for smart grid DDoS detection based on deep learning

Sayawu Yakubu Diaba *, Mohammed Elmusrati

Department of Telecommunication Engineering, School of Technology and Innovations, University of Vaasa, Vaasa, Finland

ARTICLE INFO

Article history:
Received 16 August 2022
Received in revised form 27 October 2022
Accepted 14 December 2022
Available online 21 December 2022

Keywords:
State estimation
Smart grid
Distributed denial of service
Intrusion detection
Gated recurrent unit
Convolutional neural network

ABSTRACT

The Smart Grid's objective is to increase the electric grid's dependability, security, and efficiency through extensive digital information and control technology deployment. As a result, it is necessary to apply real-time analysis and state estimation-based techniques to ensure efficient controls are implemented correctly. These systems are vulnerable to cyber-attacks, posing significant risks to the Smart Grid's overall availability due to their reliance on communication technology. Therefore, effective intrusion detection algorithms are required to mitigate such attacks. In dealing with these uncertainties, we propose a hybrid deep learning algorithm that focuses on Distributed Denial of Service attacks on the communication infrastructure of the Smart Grid. The proposed algorithm is hybridized by the Convolutional Neural Network and the Gated Recurrent Unit algorithms. Simulations are done using a benchmark cyber security dataset of the Canadian Institute of Cybersecurity Intrusion Detection System. According to the simulation results, the proposed algorithm outperforms the current intrusion detection algorithms, with an overall accuracy rate of 99.7%.

1. Introduction

The modernized grid enables a two-way flow of electricity and information while providing efficient, dependable, computerized, and decentralized energy distribution. The Supervisory Control and Data Acquisition (SCADA) Master Terminal Unit (MTU) and the Intelligent Electronic Devices (IED) on the electric network establish communication. The Remote Terminal Units (RTUs), Phasor Measurement Unit (PMU), Micro Phasor Measurement Unit (μPMU), and Programmable Logic Controls (PLC) mounted at various locations on the electric network provide telemetry data to the SCADA's server (Oyewole & Jayaweera, 2020). Electric utilities all around the world use various SCADA protocols to communicate between IEDs on the network and control center applications using different SCADA protocols, such as International Electrotechnical Commission (IEC) 61850, Modbus, and Distributed Network Protocol 3 (DNP3) (Mohan, Ravikumar, & Govindarasu, 2020). With these SCADA protocols, parameters are measured, processes are monitored, and operations are controlled using measurement and control systems (Yohanandhan, Elavarasan, Manoharan, & Mihet-Popa, 2020), which are frequently utilized in operational technology (OT) such as Smart Grid.

The SCADA system in the context of the electric network is a crucial infrastructure made up of computer-based networked systems that exchange important data across networks. Such systems are vulnerable to intrusion attacks owing to the extensive use of information technology (Liu, Li, Shuai, & Wen, 2017). Therefore, one crucial task is to evaluate the system security by considering the probable attack that could be launched by network intruders from the communication network lateral. Knowing the system security valuation would help maintain the modern electric infrastructure's security and operational stability (Fu et al., 2019).

Intrusion detection is an approach to identifying attacks before or after gaining access to a secure network. Incorporating this approach into the gateway is the quickest way to integrate it with an IEC61850-based network. Even though attack detection and self-healing are not specified in IEC 61850, a specific technique like Intrusion Detection System (IDS) may be employed within the grid to support IEC 61850's security (Elgargouri, Virrankoski, & Elmusrati, 2015). As machine-to-machine (m2 m), and human-machine-interface (HMI) connectivity increases, the potential hostile threats in the electric infrastructure become prevalent. The IDS is essential for monitoring Smart Grid security and situational awareness (Hu, Yan, & Liu, 2020; Ullah & Mahmoud, 2017). Likewise, the transmission of data via the radio medium which represents the fundamental pillar by which all devices in the Smart Grid network communicate has become prone to cyber-attack. Due to the interconnectivity (Chen, Zhang, Liu, & Tang, 2018) of the various technologies (Attia, Sedjelmaci, Senouci, & Aglzim, 2015) which was not historically known in the

* Corresponding author.
E-mail address: sdiaba@uwasa.fi (S.Y. Diaba).

https://doi.org/10.1016/j.neunet.2022.12.011

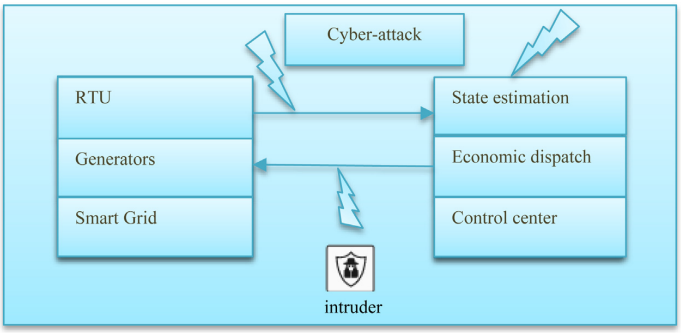


Fig. 1. Depicts a cyberattack on the smart grid.

electric networks. This makes the system vulnerable to intrusion attacks (Mahmud, Vallakati, Mukherjee, Ranganathan, & Nejadpak, 2015), which can result in significant financial losses (Gao, Li, Jiang, Li, & Quan, 2020; Jiang, Xu, Zhang, Hong, & Cai, 2020) but, more crucially, put public safety at risk. The risk is increased when new connections are added to such critical infrastructures. Therefore, a high-priority area of study in the realm of cyber security is intrusion detection in the SCADA network of a Smart Grid (Hosseinzadehtaher, Khan, Shadm, & Abu-Rub, 2020; Xu, 2020) (see Fig. 1).

On the other hand, distributed generation (DG) has been the means to shift toward renewable energy sources (RES). Establishing DG at various points of an existing network affects the primary contour of the electric network. This causes alterations in voltage and current at different nodal points and also increases the points of entry into the electric network (de Figueiredo, Ferst, & Denardin, 2019). The total Smart Grid's communication technologies and supporting infrastructure are directly impacted by the scale of the electrical network (Talha & Ray, 2016).

Looking at the shortcomings of the current Smart Grids communication mechanisms has inspired several researchers to explore cyber risks to Smart Grids. We propose an algorithm for detecting Distributed Denial of Service (DDoS) in Smart Grid in response to the aforementioned facts. The DDoS includes bombarding a target with a large volume of data and internet traffic, typically with the aid of a network of compromised machines. The following summarizes this paper:

- To identify DDoS attacks we propose an algorithm hybridized by a Convolutional Neural Network (CNN) and a Gated Recurrent Unit (GRU) for DDoS attacks in the cyber–physical system of the Smart Grid.
- Utilizing benchmark datasets from the Canadian Institute of Cybersecurity Intrusion Detection System (CICIDS2017), in-depth simulation studies are presented. Comparative analyses are drawn and the proposed algorithm performed better in comparison to other state-of-the-art algorithms with a 99.7% accuracy and 99.9% detection rate.

The remainder of the paper is organized as follows. A review of the literature is in Section 2. Presenting the proposed hybrid algorithm is given in Section 3. The proposed algorithm's performance is compared to current algorithms in simulations described in Section 4. Finally, concluding observations are made in Section 5.

2. Related studies

Communication networks' reliability, confidentiality, and integrity are just a few of the difficulties involved in protecting sensitive infrastructure, such as Smart Grid. To protect this crucial infrastructure, the Smart Grid requires a security strategy. It is necessary to meet requirements for data authentication, confidentiality and integrity assurance, and other security-related issues (Subasi et al., 2018). Owing to the above-stated reasons, researchers have evaluated intrusion detection in the cyber–physical of the Smart Grid from different perspectives. For example, Li et al. suggested various monitoring measures to track suspicious branch flow changes and abnormal load deviations. Two-stage approaches are suggested to identify false data injection (FDI) attacks. The article introduces the FDI cyber-attack to investigate the impact of FDI attacks on system reliability (Li & Hedman, 2020). The alert system with the developed unique metrics serves as the foundation for the suggested FDI detection approach.

A customized firewall model SCADAWall was proposed to address the limitations of the traditional firewall system in protecting the SCADA networks (Li, Guo, Zhou, Zhou, & Wong, 2019). The traditional SCADA systems were working in the principle of deep packet inspection that was designed to inspect the payload contents in the communication. A proprietary industrial protocols extension algorithm and an out-of-sequence detection algorithm were added to the SCADAWall to improve its ability to identify abnormal changes in industrial operations. The experimental analysis indicates that the SCADAWall framework is effective in the detection process by maintaining the latency parameters of the SCADA system (Li et al., 2019). A testbed model was developed for SCADA systems (Almgren, 2018) to confirm the effectiveness of the suggested algorithms in a real-time scenario. The virtual model is equipped with an energy management model monitored by a SCADA system. The testbed was created to give various real-world scenarios like attack generation and defense algorithms. An anomaly-based method was created to detect malicious packet movement in the SCADA network (Singh, Ebrahem, & Govindarasu, 2018). The experimental work indicates a better latency and detection rate. The rule-based intrusion detection system presented in Yang et al. (2013) employs a deep packet inspection technique and was designed specifically for SCADA systems. It also contains signature-based and model-based techniques. The suggested signature-based rules are capable of correctly identifying several known suspicious or malicious assaults.

An algorithm was made to address the SCADA system's Dynamic Link Library (DLL) injection attack (Lee & Hong, 2020). The model utilizes the Windows Application Programming Interface (API) function that verifies the changes in the DLL load and enables the diversion algorithm when an attack is detected. A security layer was structured between the physical and link layer of the SCADA system to overcome the issues observed from the existing firewall and authentication mechanisms (Cherifi &

Hamami, 2018). An IEC 60870 — 5 — 101 communication protocol was employed in the work and that is un-routable by the intrusion algorithms. The simulation implementation of the security layer protection in an electrical substation testbed indicates a satisfactory performance over the previous models.

An analysis was performed to identify the effectiveness of artificial intelligence (AI)-based techniques in detecting Denial of Service (DoS) attacks in SCADA systems (Aldossary, Ali, & Alasaadi, 2021). The experimental result indicates that a model developed as Bidirectional Long Short-Term Memory (Bi-LSTM) was capable of detecting intrusions against the other methods. To identify intrusion detection and DoS in the smart meter, a cyber–physical monitoring system was proposed (Sun, Guan, Liu, & Liu, 2013). The idea is predicated on the informational fusion of online occurrences and objective data. The test shows that by linking the cyber and physical signals, the model successfully detects threats.

A temporal pattern recognition technique was proposed to observe the cyber-attack intrusions in the SCADA systems (Kalech, 2019). The technique was also designed to monitor the abnormal changes in the operation of the connected system. This was achieved by implementing the model with a hidden Markov model and the artificial neural network (ANN) algorithm. The effectiveness of the proposed model was verified with simulations and real-time scenarios with five different feature extraction strategies and the approach that was implemented with the time feature extraction model was found satisfactory.

A C4.5 decision tree algorithm was proposed to give a security model over the SCADA system implemented in gas and oil plants. The performance analysis of the proposed model explores a betterment in handling large-scale distributed attacks in the SCADA setup (Yang, Liu, & Zhang, 2019). A SCADA network attack detection technique was developed with a random forest algorithm and its attainments were compared over the support vector machine (SVM). It indicates a 96.47% of f1 score on detecting the DoS attacks (Lopez Perez, Adamsky, Soua, & Engel, 2018). The performances of the decision tree and K-nearest neighbor algorithms (KNN) were analyzed on cyber security identification. The experimental work was performed with three different cybersecurity datasets. The work findings found satisfactory results with a fine tree and weighted KNN (Ahakonye, Nwakanma, Lee, & Kim, 2021). A DDoS attack detection approach on the SCADA system was performed with J48, Naïve Bayes, and random forest algorithms. The experimental work utilizes the KDDCUP99 dataset for the analysis and was found satisfied with the accuracy rate of 99.99% in the random forest algorithm (Alhaidari & AL-Dahasi, 2019).

For Software Defined Networking (SDN), the authors of Fouladi, Ermiş, and Anarim (2022) provided a DDoS attack detection and countermeasure technique based on discrete wavelet transform and auto-encoder neural network. In the suggested method, wavelet transform was used to extract statistical features that are then processed by an auto-encoder neural network to identify samples of DDoS attacks. In order to effectively resist DDoS attacks, a novel feature selection-whale optimization algorithm deep neural network approach is presented in Agarwal, Khari, and Singh (2021). The usual data are homomorphically encrypted and safely stored in the cloud to increase the security of the proposed paradigm. A 95.35% accuracy in detecting DDoS attacks was shown by simulation results. A swarm intelligence technique was developed to identify the optimum features for making a good accuracy rate in the intrusion detection system process. An Aquila optimizer model was also employed in the work after the feature selection process for assigning desirable weights to the extracted features. The work offered a reasonable result when implemented with a CNN classifier with a particle swarm optimization model (Fatani, Dahou, Al-qaness, Lu, & Abd Elaziz, 2022).

Concerning internet-based computer network attacks, a neural network-based intrusion detection method is presented in Shum and Malki (2008). IDS were developed to foresee and stop potential attacks. To find and forecast anomalous system behavior, neural networks were used. The study specifically used feedforward neural networks with the back-propagation training algorithm. The experimental outcomes utilizing real data demonstrated positive outcomes for neural-network-based IDS. In Peng, Kong, Peng, Li, and Wang (2019), a deep learning-based technique for network intrusion detection is presented. In the model, network monitoring data features are extracted using deep neural networks, and intrusion types are classified at the top-level using back propagation neural networks. The KDDCUP99 dataset from the Massachusetts Institute of Technology's Lincoln Laboratory was used to validate the approach. The findings indicate that the proposed method meaningfully outperforms the accuracy of conventional machine learning. In Hai-He (2018) the authors proposed an IDS based on the improved neural network where feature extracting was carried out using the adaptive weighted control method. The model showed higher accuracy using a back propagation neural network for classification and detection. However, the back propagation neural network algorithm proposed in Jaiganesh, Sumathi, and Mangayarkarasi (2013) with the primary duty of detecting threats to the resources demonstrated a poor attack detection rate.

To categorize network threats, the study in Lin, Lin, Wang, Wu, and Tsai (2018) concentrated on network intrusion detection utilizing CNNs based on LeNet-5. The experiment's findings indicate that with samples larger than 10,000, intrusion detection prediction accuracy increases and gains overall accuracy of 97.53%. The authors of Khan, Zhang, Alazab, and Kumar (2019) offer a network intrusion detection approach using CNN. The approach is intended to efficiently categorize intrusion data by automatically extracting useful features from intrusion samples. An automated vision-based android malware detection algorithm was proposed with a fine-tuned CNN algorithm. The byte codes extracted from the various malware devices are collected in the work for training the classifiers. The experimental work attains an accuracy of 99.4% and 98.05% on both balance and imbalanced datasets (Almomani, Alkhayer, & El-Shafai, 2022). In the blockchain-based energy network, Ferrag and Maglaras (2019) presented a learning-based method to identify network threats and fraudulent transactions. The suggested system generates blocks using short signatures and hash functions to thwart Smart Grid attacks.

Peng (2020) propose a hybrid CNN-based intrusion detection approach. The hybrid deep learning network structure extracts and encapsulates the features of unfamiliar malicious behavior as well as more complex structure aspects of the full network traffic matrix, in contrast to the typical machine learning approach. In the network traffic matrix, a CNN first extracts the correlation between several features. Then, by using a Recurrent Neural Network (RNN) to fully mine the temporal and spatial features of the entire network traffic matrix, the accuracy of the intrusion detection model is boosted. Al-Emadi, Al-Mohannadi, and Al-Senaid (2020) developed an intelligent detection system that can recognize various network intrusions using deep learning approaches, specifically CNN and RNN. The authors compared the results of the offered solution and evaluated the performance of the proposed solution using several evaluation matrices to select the best model for the network IDS. Koutsandria et al. suggested a hybrid control paradigm that constantly tracks and examines the network traffic that is transferred inside the physical system. It detects communication patterns that diverge from expectations or physical constraints that can put the system in a dangerous mode of operation. The simulations show that, by utilizing data

on the physical component of the power system, the paradigm is capable of identifying a wide variety of attack scenarios intended to compromise the physical process (Koutsandria et al., 2014).

In Vijayanand, Devaraj, and Kannapiran (2019) a unique attack detection system that uses deep learning algorithms to detect attacks by carefully examining smart meter communications is presented. To detect cyber-attacks accurately, the attack detection system uses several multi-layer deep algorithms that are set up in a hierarchical order. In Farrukh, Ahmad, Khan, and Elavarasan (2021) the authors proposed a two-layer hierarchical machine learning model with 95.44% accuracy in detecting cyber-attacks. Using the model's first layer, the two modes of operation, normal state and cyberattack are identified. The authors of Zhao, Chen, and Luo (2011) suggested a methodology incorporating real-time neural network training and expert system detection to improve detection accuracy. The model employs neural networks to detect and converts pattern recognition into numerical calculation to speed up the detection rate. The state is divided into many categories of cyberattacks using the second layer.

In our humble opinion, as so many articles have consisted of IDS in power systems annals with little reference to the hybridization milieu, a revisit of that background could yield a novelty. This paper seeks to present one.

3. System model

Fig. 2 shows the proposed hybrid deep learning algorithm. In our earlier study (Diaba, Shafie-khah, & Elmusrati, 2022), this algorithm was tested using the Network Security Laboratory-Knowledge Discovery and Data Mining (NSL-KDD99) dataset, and the results were compared with CNN, GRU, and LSTM algorithms. The algorithm performed better in terms of accuracy, detection rate, precision, and force positive rate (FPR). However, Elmrabit, Zhou, Li, and Zhou (2020) argued that the NSL-KDD99 dataset had expired. Since the network traffic in that dataset was established in 1998, the authors claimed that it is impossible for it to accurately reflect the most recent network topologies and attack dynamics. We, therefore, seek to apply the CICIDS2017 cyber security dataset to the algorithm because of the presence of a large variety of up-to-date attack scenarios in the dataset, which satisfy real-world requirements.

The proposed IDS integrates a CNN model and a GRU model. It is believed that CNN is effective at capturing position-invariant characteristics, thus the choice. The GRU module collects the long-dependence features and uses memory cells to extract key information from the previous data. The reset gate is employed to erase or eliminate pointless data. These influenced the decision to use the GRU model (Aldossary et al., 2021). Three GRU blocks and four CNN blocks are mounted in the algorithmic architecture to deepen the network (Huang, Li, Deng, Yu, & Ma, 2022). The purpose of the convolution layer is to produce a feature map by separating features from the input data. To capture the feature mapping, the input data are multiplied by the convolutional kernel in the convolutional network, which is then activated by a nonlinear function. The convolution kernel randomly initializes weights and biases (Liang, Ye, Zhou, & Yang, 2021). After each CNN layer, a normalization layer and a max-pooling layer are added. The procedure of obtaining the maximum or average value for all features within the immediate area is referred to as a "pooling operation".

The concatenation layer, where the GRUs output and the CNN outputs are combined, receives the flattened final output of the CNN layers. Two completely connected layers are connected after the concatenation layer. A dropout layer is used after the last fully connected layer to prevent overfitting. The SoftMax layer connects to the classification layer to map the output to a probability distribution, which allows the classification layer to predict the types of labels.

3.1. Deep neural network structure

Artificial neural networks were inspired by research on biological neural network processing, a type of computer structure. An artificial neural network is a self-motivated system made up of highly connected, parallel nonlinear processing components, units, or nodes that exhibit extremely high levels of computation efficiency. It can alternatively be viewed as versatile mathematical structures that can recognize intricate nonlinear correlations between input and output datasets (Suppitaksakul & Saelee, 2009). A typical neural network comprises numerous small, interconnected processes called neurons, each generating a string of activations with real values. Environmental sensors activate input neurons, and weighted connections from previously active neurons excite more neurons (Komyakov, Erbes, & Ivanchenko, 2015; Liang et al., 2021; Schmidhuber, 2015) (see Fig. 3).

3.2. System description

The mathematical formulation of the proposed algorithm considers a features vector ξ, given as

$$\xi = [\xi_1, \xi_2, \ldots, \xi_n]^T \tag{1}$$

as inputs to the proposed model. The GRU's first layer processes the data and generates the outputs. The first layer's outputs are fed into the second layer. Again, the outputs of the second layer are inputted into the third layer. The final outputs of the GRU model are achieved by using an activation function. We apply the most used activation functions, the sigmoid, and the tanh, respectively, given as in Ismail et al. (2022) and Valdes, Macwan, and Backes (2016).

$$s(x) = \frac{1}{1 + e^{1-x}} \tag{2}$$

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{3}$$

Mathematically, the GRU gate $\in\{0,1\}$ and thus, the model can be written mathematically as

$$\varsigma_0 = \sigma\left(\omega_0\left[\widetilde{\xi}^{(t-1)}, x^{(t)}\right] + b_0\right) \tag{4}$$

$$\varsigma_1 = \sigma\left(\omega_1\left[\widetilde{\xi}^{(t-1)}, x^{(t)}\right] + b_1\right) \tag{5}$$

where, ς_0, ς_1 represent the update gate and the reset gate, respectively. The ω_0 and ω_1 are weights functions representing the update and reset gates in the order given. Correspondingly, the b_0 and b_1 represents the bias vectors for reset and update gates. Where $\widetilde{\xi}^{(t-1)}$ is the input of the current layer and the output of the prior layer. The recurrent unit's candidate activation function is written as

$$\widetilde{\xi}^{(t)} = tanh\left(\omega_0\left[\varsigma_0 \times \widetilde{\xi}^{(t-1)}, x^{(t)}\right] + b_0\right) \tag{6}$$

where, $\tilde{\xi}^{(t)}$ represents the candidate activation function, ω_0 is the activation functions weight, the bias vector is b_0 and $x^{(t)}$ is the inputs of the training data. One GRU unit's output is provided as

$$\xi^{(t)} = \left((1 - \zeta_1) \times \widetilde{\xi}^{(t-1)}\right) + \left(\zeta_1 \times \xi^{(t)}\right) \tag{7}$$

where, $\xi^{(t)}$ is the output of a single GRU unit. The proposed algorithm uses a single-dimensional layer with the convolution operation represented as

$$h_1 = convblock1\,(\xi) \tag{8}$$

$$h_2 = convblock2\,(h_1) \tag{9}$$

$$h_3 = convblock3\,(h_2) \tag{10}$$

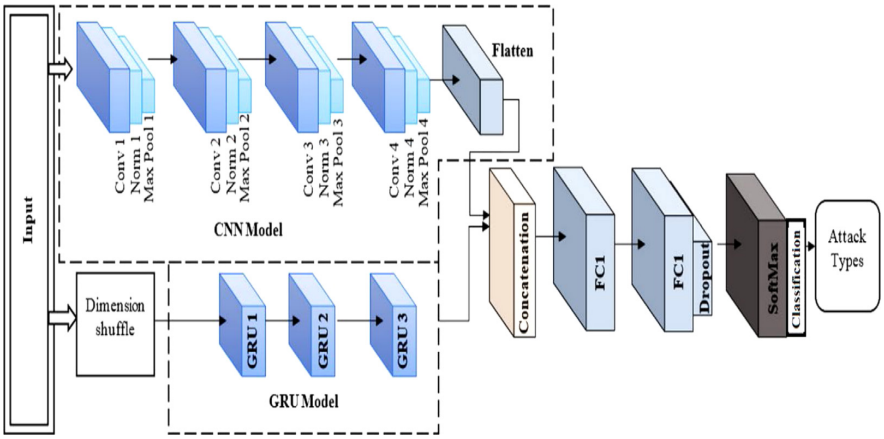


Fig. 2. Process flow of proposed intrusion-detection system model.

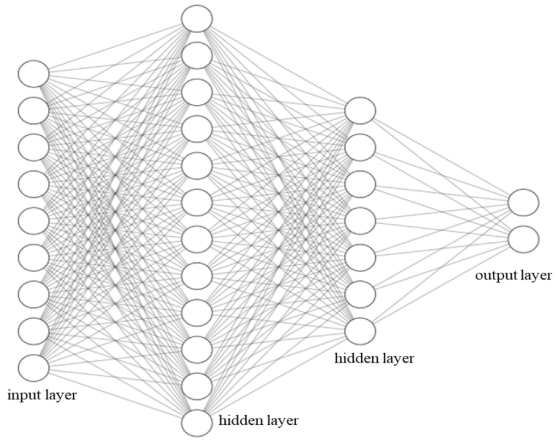


Fig. 3. Illustrations of a feed-forward multilayer perceptron.

$$h_4 = convblock4\,(h_3) \tag{11}$$

The hidden vectors are h_1, h_2, h_3, and h_4 respectively. A normalization layer is fixed next to the convolutional layer to speed up training. By using the pooling layer, the features map is down-sampled by summarizing the presence of features in patches of the feature map, hence reducing the dimension of the features. The main pooling techniques are average and max pooling. The average pooling determines the average value of the patches of the features map. The average pooling at the pooling layer is given as

$$\Psi^n = P_{avg}\ \left(\psi^{n-1}\right) \tag{12}$$

where ψ^{n} represents the pooling layers, and output and $\psi^{\mathrm{n-1}}$ represent previously acquired values from the convolution layer. The pooling layers are denoted by n and the flattening layer is mounted to convert the data into a one-dimensional vector.

$$L = flatten\,(h_4) \tag{13}$$

$$c_t = concate\,(K, L) \tag{14}$$

Table 1
Dataset considered for the simulation.

Type	Total	Training set	Test set	Label
BENIGN	67,343	53874	13469	0
DDoS	45,927	36742	9185	1

The outputs from the GRU's K, and the outputs from the CNN's L, are concatenated as written in Eq. (14). The normalized exponential function (SoftMax) $\hat{y}: \mathbb{R}^{c_t} \rightarrow \{0, 1\}$ is written when c_t is greater than 1 as

$$\hat{y}(z)_i = \frac{e^{z_i}}{\sum_{n=1}^{c_t} e^{z_n}} \tag{15}$$

For $i = 1, 2,\ldots, c_{\mathrm{t}}$ and $\mathbf{z} = (z_1, z_2,\ldots,zc_{\mathrm{t}}) \in \mathbb{R}^{c_t}$. Where $\mathbf{z}$ is the input vector taken from the c_t. The loss function for the proposed model assessment is the cross-entropy function (Graves & Schmidhuber, 2005), which is given as

$$E_p\ (l) = -\frac{1}{b}\sum_{i=1}^{n} y_i\, log_2\, y_i' \tag{16}$$

b is for the batch size given, whilst n represents the training sample size, the actual value is represented by y_i and it is y_i' for the predicted value.

3.3. Description of dataset

The simulation evaluation phase of our proposed model is carried out using the CICIDS-2017 (Radoglou-Grammatikis & Sarigiannidis, 2018) dataset, specifically, the *Friday WorkingHours Afternoon DDos* dataset (Sharafaldin, Habibi, & Ghorbani, 2018) which is publicly accessible and utilized by related studies in the cyber security community. The *benign* and most recent common attacks such as *DDoS* are included in the CICIDS-2017 dataset, which closely reflects data from the actual world. Additionally, it contains the outcomes of the CICFlowMeter network traffic analysis with flows categorized according to the source, timestamp, destination IP addresses, destination ports, protocols, and attacks. The features present there in the dataset are shown in Table 1.

Cleaning up the data and replacing not a number (NaN) and infinite fields with the column's mean value are the first steps in the preprocessing stage. The features are converted to numerical features and integrated with already-existing numerical features

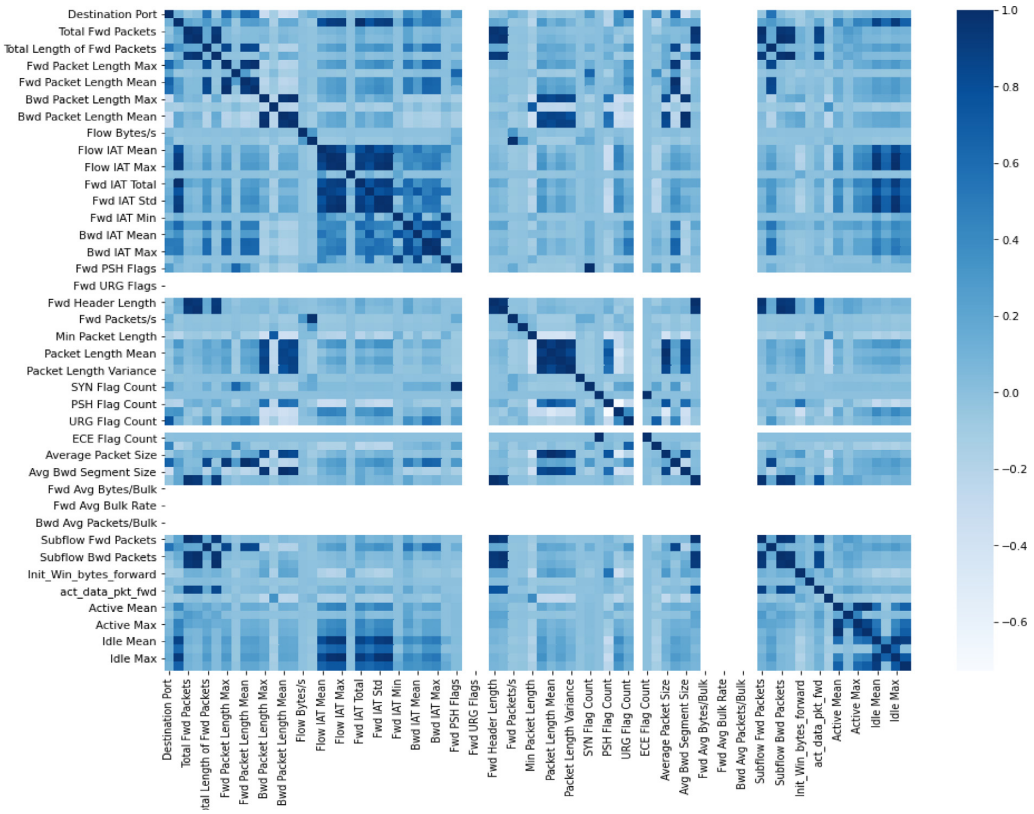


Fig. 4. The correlation heatmap for the employed dataset.

in the dataset. Additionally, the labels in the dataset are numerically processed so that the two labels in the dataset, *benign* is represented by 0, and *DDoS* is represented by 1. The dataset is equally mapped and normalized in order to lessen the feature discrepancies. The uniform mapping interval range is [0, 1]. Since there are no irrelevant characteristics in the dataset and the dataset contains correlated features as shown in the correlation matrix in Fig. 4, feature selection was not used in the study. Therefore, the model's decision-making was influenced by all of the available features.

The list in Table 1 is the results after the normalization had been performed on the data set and therefore all the character features had been converted to their numerical values. Then, the data set is split into a training set and a testing set in a 70:30 ratio. The training is done using 70 percent of the data, and the validation and testing are done using the remaining 30 percent of the data.

The four fundamental characters that make up the confusion matrix are utilized to specify the classifier's measurement parameters. They are as follows: True Positive (TP) describes an algorithm's accurate prediction that is accurate. Also, the True Negative (TN) designates a truly negative prediction made by the algorithm that is negative. False Positive (FP) describes situations where the algorithm predicted a positive class but the actual class is negative. False Negative (FN) is a label that was predicted by the algorithm to be negative but is actually positive. An algorithm's performance measurements are its accuracy, precision, recall, and f1-score. These scenarios are mathematically represented as in Albulayhi and Sheldon (2021), Khoei, Aissou, Hu, and Kaabouch (2021), Peng et al. (2019), Radoglou-Grammatikis and Sarigiannidis (2018), Sharafaldin et al. (2018) and Siniosoglou, Radoglou-Grammatikis, Efstathopoulos, Fouliras, and Sarigiannidis (2021) and written in subsequent equations as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

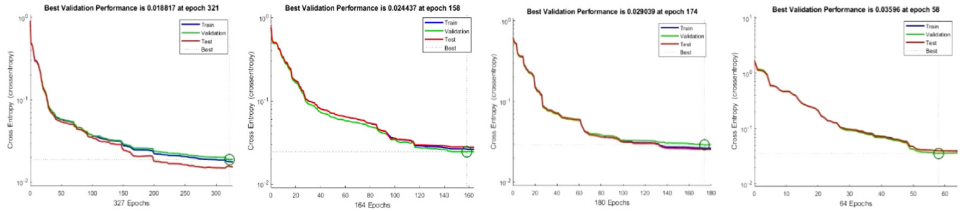
$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$F1score = \frac{2(precision \times recall)}{(precision \times recall)} \tag{20}$$

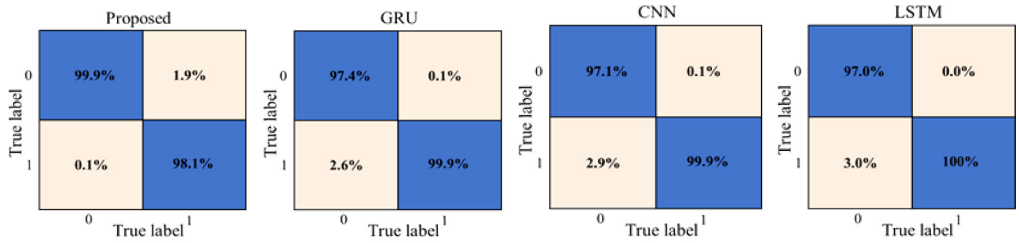
4. Results and analysis

The simulation results of our proposed algorithm are all contained in this section. Figures representing each outcome are presented step-by-step along with explanations of the findings. We give a succinct explanation of our proposed algorithm's performance and comparisons to that of some of its main contestants such as CNN, GRU, and LSTM. The heatmap depicts the correlation matrix between the target variable and the input features, including the destination port, flow bytes, forward header length, subflow forward packet, active mean, minimum packet length, packet length mean, packet length variance, average packet size, active max, ideal mean, ideal max, etc.



a. Proposed algorithm. **b.** GRU. **c.** CNN. **d.** LSTM

Fig. 5. Convergence ability of the considered algorithms.



a. Proposed algorithm. **b.** GRU. **c.** CNN. **d.** LSTM

Fig. 6. The confusion matrices..

Table 2
The configuration of the hyperparameters.

Number	Parametric	Quantity
1.	Input layer	78
2.	Hidden layer	55
3.	Activation function	ReLU
4.	Iteration limit	1000
5.	Cost function	Cross entropy
6.	Batch size	128

A heat map is a graphic representation of a two-dimensional tabular representation of multivariate data that is set up as a matrix. The heat map shows the relationships between several numerical variables, which can be used to identify patterns and anomalies. It helps to find characteristics that are best for developing machine learning models and transforms the correlation matrix into a color designation. It generates color coding from the correlation matrix and the correlation matrix shows the relationships between the variables on a scale from a perfect positive correlation to a perfect negative correlation with the perfect positive correlation showing the association between the variables. Each cell represents a square region of space in a certain measuring distance, and the colors signify the intensity of the investigated event that occurred on each mapping cell. A heat map provides a visual representation of data and facilitates the understanding of large data sets. A range of values is represented by various colors in a two-dimensional tabular depiction of the data.

Further simulations are run with the hyperparameters settings in Table 2. The proposed algorithm's convergence ability outperforms that of the other comparative algorithms. The algorithm's best validation performance is achieved at 0.018817 at epoch 321. The GRU is the next best-performing algorithm, with the best validation performance at 0.024437 on the 158th epoch. The CNN algorithm also outperformed the LSTM algorithm, achieving the best validation performance of 0.029039 at the 174th epoch, while the LSTM achieved its best validation performance of 0.03596 at an epoch of 58 (see Fig. 5).

The confusion matrices of the performance of the algorithms are depicted in Fig. 6a, b, c, and d. A confusion matrix is used to evaluate the algorithms based on parameters such as accuracy, precision, recall, and the false positive rate (Aldossary et al., 2021).

The error histograms are depicted in Fig. 7 to determine the error between the predicted and target values. Bins are the vertical bars seen on the graph. The total error range is divided into 20 smaller bins on the x-axis. The Y-axis represents the number of samples from the input dataset that fall into a given bin. On the plot, the midpoint bin corresponds to an error of 0.01599, the height of the bin for the training dataset is below 2×10^4 and the height of the bin for validation is between a little below 2×10^4 and halfway above 2×10^4. The test dataset is halfway between 2.5×10^4 and 2×10^5.

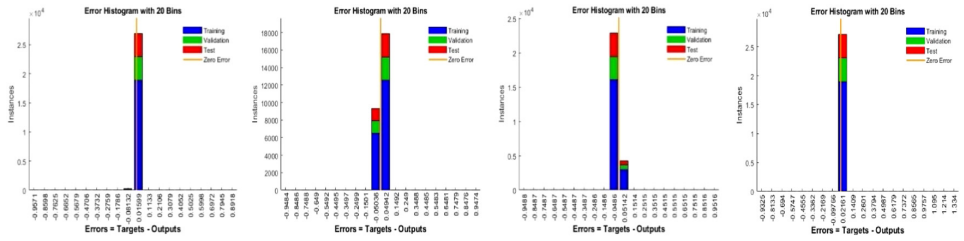
In terms of overall accuracy, precision, recall, and f1-score, Fig. 8 shows how well the proposed algorithm performed against the other algorithms Simulation results show that the proposed algorithm achieves accuracy, precision, recall, and f1-score, of 99.7%, 98.1%, 99.9%, and 98.9%, respectively. The GRU achieves an accuracy of 98.6%, precision of 99.5%, recall of 97.4%, and an f1-score of 98.5. The accuracy of the CNN is 98.5%, the precision is 99.8, the recall is 97.3% and the f1-score is 98.5%. The LSTM obtains 98.5% accuracy, 99.9% precision, 97% recall, and an f1-score of 98% FPR. The proposed model outperformed the comparative algorithms in all categories except the recall category. This is a result of the algorithm's high value of the false positive (FP). Since the FP is a denominative factor in determining the recall, its higher value caused the recall of the proposed algorithm to drop (see Table 3).

5. Conclusion

Finding vulnerabilities in SCADA networks used by Smart Grids is a top research objective in the field of cyber security. However, it is very challenging to choose an efficient deep

Table 3
Comparison of algorithms.

The proposed algorithm is compared to the existing algorithms altogether							
Algorithms	Detection rate %	Precision %	F1-score	Accuracy %	Data	Year	Reference
ANN	96.18		96.9	96.94	Simulated data	2018	Subasi et al. (2018)
SVM	97.25		97.8	97.8	Simulated data	2018	Subasi et al. (2018)
K-NN	98.05		98.4	98.44	Simulated data	2018	Subasi et al. (2018)
Randon forest	98.67		0.98	98.94	Simulated data	2018	Subasi et al. (2018)
Feed-forward neural network	90.13	88	87.4	88.2	Power system attack	2021	Aldossary et al. (2021)
Hybrid Deep belief network GRU	93.5	93.57	93.68	94.14	Power system attack	2021	Aldossary et al. (2021)
Recommended Bi-LSTMIDS	99.89	95.89	95.94	95.93	Power system attack	2021	Aldossary et al. (2021)
Random forest				99.9	KDDCup'99	2019	Alhaidari and AL-Dahasi (2019)
Naïve Bayes				97.74	KDDCup'99	2019	Alhaidari and AL-Dahasi (2019)
Proposed scheme	100	99.9	99.9	99.9	MAWI and world cup traffic dataset	2022	Fouladi et al. (2022)
Random forest	94			94	CICDDoS 2019	2021	Khoei et al. (2021)
Naïve Bayes	87			77.1	CICDDoS 2019	2021	Khoei et al. (2021)
KNN	94.4			94.6	CICDDoS 2019	2021	Khoei et al. (2021)
Stacking	96			97.3	CICDDoS 2019	2021	Khoei et al. (2021)
Logistic regression	72.2		72.2	90.7	Distribution substation operational dataset	2021	Siniosoglou et al. (2021)
Decision tree	99.1		99.1	97.7	Distribution substation operational dataset	2021	Siniosoglou et al. (2021)
Multi-layer perceptron	73.3		73.3	91.1	Distribution substation operational dataset	2021	Siniosoglou et al. (2021)
Proposed algorithm	99.9	98.1	98.9	99.7	CICIDSS2017	2022	



a. Proposed algorithms. **b.** GRU. **c.** CNN **d.** LSTM

Fig. 7. Error histograms.

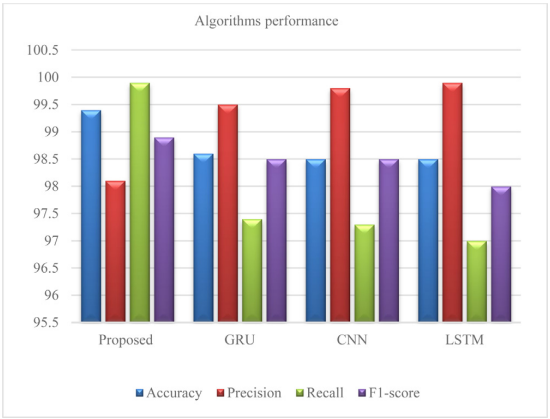


Fig. 8. Overall performance comparison of the considered algorithms.

learning-based intrusion detection algorithm. As a result, we proposed an algorithm for intrusion detection in Smart Grid, by hybridizing CNN and GRU algorithms. In evaluating the efficacy of our proposed algorithm, the accuracy, precision, recall, and f1-score, are evaluated to strengthen the SCADA system's security framework and make it more resistant to DDoS attacks. Using the CICIDSS2017 dataset, we carried out a thorough systematic simulation using MATLAB 2021a. We used the supervised machine learning approach after normalizing the data. Results demonstrate that the proposed algorithm can classify cyberattacks with a 99.7% accuracy and a detection rate of 99.9%, outperforming the accuracy and the detection rate of the comparative existing intrusion detection techniques. In general, the proposed algorithm can improve network intrusion detection performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Agarwal, A., Khari, M., & Singh, R. (2021). Detection of DDOS attack using deep learning model in cloud storage application. *Wireless Personal Communication*, http://dx.doi.org/10.1007/s11277-021-08271-z.

Ahakonye, L. A. C., Nwakanma, C. I., Lee, J. M., & Kim, D. S. (2021). Efficient classification of enciphered SCADA network traffic in smart factory using decision tree algorithm. *IEEE Access*, *9*, 154892–154901. http://dx.doi.org/10.1109/ACCESS.2021.3127560.

Al-Emadi, S., Al-Mohannadi, A., & Al-Senaid, F. (2020). Using deep learning techniques for network intrusion detection. In *2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT)* (pp. 171–176). http://dx.doi.org/10.1109/ICIoT48696.2020.9089524.

Albulayhi, K., & Sheldon, F. T. (2021). An adaptive deep-ensemble anomaly-based intrusion detection system for the internet of things. http://dx.doi.org/10.1109/AIIoT52608.2021.9454168, 0187-0196.

Aldossary, L. A., Ali, M., & Alasaadi, A. (2021). Securing SCADA systems against cyber-attacks using artificial intelligence. In *2021 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT)* (pp. 739–745). http://dx.doi.org/10.1109/3ICT53449.2021.9581394.

Alhaidari, F. A., & AL-Dahasi, E. M. (2019). New approach to determine ddos attack patterns on SCADA system using machine learning. In *2019 international conference on computer and information sciences* (pp. 1–6). http://dx.doi.org/10.1109/ICCISci.2019.8716432.

Almgren, M. (2018). Building a national testbed for research and training on SCADA security (short paper). In *13th international conference, CRITIS 2018, Kaunas, Lithuania*. Springer.

Almomani, I., Alkhayer, A., & El-Shafai, W. (2022). An automated vision-based deep learning model for efficient detection of android malware attacks. *IEEE Access*, *10*, 2700–2720. http://dx.doi.org/10.1109/ACCESS.2022.3140341.

Attia, M., Sedjelmaci, H., Senouci, S. M., & Aglzim, E.-H. (2015). A new intrusion detection approach against lethal attacks in the smart grid: temporal and spatial based detections. In *2015 global information infrastructure and networking symposium* (pp. 1–3). http://dx.doi.org/10.1109/GIIS.2015.7347186.

Chen, X., Zhang, L., Liu, Y., & Tang, C. (2018). Ensemble learning methods for power system cyber-attack detection. In *2018 IEEE 3rd international conference on cloud computing and big data analysis* (pp. 613–616). http://dx.doi.org/10.1109/ICCCBDA.2018.8386588.

Cherifi, T., & Hamami, L. (2018). A practical implementation of unconditional security for the IEC 60780 – 5 – 101 SCADA protocol. *International Journal of Critical Infrastructure Protection*, *20*, 68–84.

de Figueiredo, H. F. M., Ferst, M. K., & Denardin, G. W. (2019). An overview about detection of cyber-attacks on power SCADA systems. In *2019 IEEE 15th Brazilian power electronics conference and 5th IEEE southern power electronics conference (COBEP/SPEC)* (pp. 1–6). http://dx.doi.org/10.1109/COBEP/SPEC44138.2019.9065353.

Diaba, S. Y., Shafie-khah, M., & Elmusrati, M. (2022). On the performance metrics for cyber–physical attack detection in smart grid. *Soft Computing*, http://dx.doi.org/10.1007/s00500-022-06761-1.

Elgargouri, A., Virrankoski, R., & Elmusrati, M. (2015). IEC 61850 based smart grid security. In *2015 IEEE international conference on industrial technology* (pp. 2461–2465). http://dx.doi.org/10.1109/ICIT.2015.7125460.

Elmrabit, N., Zhou, F., Li, F., & Zhou, H. (2020). Evaluation of machine learning algorithms for anomaly detection. In *2020 international conference on cyber security and protection of digital services (cyber security)* (pp. 1–8). http://dx.doi.org/10.1109/CyberSecurity49315.2020.9138871.

Farrukh, Y. A., Ahmad, Z., Khan, I., & Elavarasan, R. M. (2021). A sequential supervised machine learning approach for cyber attack detection in a smart grid system. In *2021 north American power symposium* (pp. 1–6). http://dx.doi.org/10.1109/NAPS52732.2021.9654767.

Fatani, A., Dahou, A., Al-qaness, M. A. A., Lu, S., & Abd Elaziz, M. (2022). Advanced feature extraction and selection approach using deep learning and aquila optimizer for IoT intrusion detection system. *Sensors*, *22*, 140. http://dx.doi.org/10.3390/s22010140.

Ferrag, M. A., & Maglaras, L. (2019). DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids. *IEEE Transactions on Engineering Management*, *67*(4), 1285–1297.

Fouladi, R. F., Ermiş, O., & Anarim, E. (2022). A ddos attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN. *Computer Networks*, *214*, Article 109140.

Fu, R., Huang, X., Xue, Y., Wu, Y., Tang, Y., & Yue, D. (2019). Security assessment for cyber physical distribution power system under intrusion attacks. *IEEE Access*, *7*, 75615–75628. http://dx.doi.org/10.1109/ACCESS.2018.2855752.

Gao, J., Li, J., Jiang, H., Li, Y., & Quan, H. (2020). A new detection approach against attack/intrusion in measurement and control system with fins protocol. In *2020 Chinese automation congress* (pp. 3691–3696). http://dx.doi.org/10.1109/CAC51589.2020.9327136.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*, 5–6.

Hai-He, T. (2018). Intrusion detection method based on improved neural network. In *2018 international conference on smart grid and electrical automation* (pp. 151–154). http://dx.doi.org/10.1109/ICSGEA.2018.00045.

Hosseinzadehtaher, M., Khan, A., Shadm, M. B., & Abu-Rub, H. (2020). Anomaly detection in distribution power system based on a condition monitoring vector and ultra- short demand forecasting. In *2020 IEEE CyberPELS (CyberPELS)* (pp. 1–6). http://dx.doi.org/10.1109/CyberPELS49534.2020.9311534.

Hu, C., Yan, J., & Liu, X. (2020). Adaptive feature boosting of multi-sourced deep autoencoders for smart grid intrusion detection. In *2020 IEEE power & energy society general meeting* (pp. 1–5). http://dx.doi.org/10.1109/PESGM41954.2020.9281934.

Huang, K., Li, S., Deng, W., Yu, Z., & Ma, L. (2022). Structure inference of networked system with the synergy of deep residual network and fully connected layer network. *Neural Networks*, *145*.

Ismail, et al. (2022). A machine learning-based classification and prediction technique for DDoS attacks. *IEEE Access*, *10*, 21443–21454. http://dx.doi.org/10.1109/ACCESS.2022.3152577.

Jaiganesh, V., Sumathi, P., & Mangayarkarasi, S. (2013). An analysis of intrusion detection system using back propagation neural network. In *2013 international conference on information communication and embedded systems* (pp. 232–236). http://dx.doi.org/10.1109/ICICES.2013.6508202.

Jiang, Y., Xu, A., Zhang, Y., Hong, C., & Cai, X. (2020). Anticipate fault sets generation methods for cyber physical power system considering cyber-attacks. In *2020 12th IEEE PES Asia-Pacific power and energy engineering conference* (pp. 1–5). http://dx.doi.org/10.1109/APPEEC48164.2020.9220404.

Kalech, M. (2019). Cyber-attack detection in SCADA systems using temporal pattern recognition techniques. *Computers & Security*, *84*, 225–238.

Khan, R. U., Zhang, X., Alazab, M., & Kumar, R. (2019). An improved convolutional neural network model for intrusion detection in networks. In *2019 cybersecurity and cyberforensics conference* (pp. 74–77). http://dx.doi.org/10.1109/CCC.2019.000-6.

Khoei, T. T., Aissou, G., Hu, W. C., & Kaabouch, N. (2021). Ensemble learning methods for anomaly intrusion detection system in smart grid. In *2021 IEEE international conference on electro information technology* (pp. 129–135). IEEE.

Komyakov, A. A., Erbes, V. V., & Ivanchenko, V. I. (2015). Application of artificial neural networks for electric load forecasting on railway transport. In *2015 IEEE 15th international conference on environment and electrical engineering* (pp. 43–46). http://dx.doi.org/10.1109/EEEIC.2015.7165296.

Koutsandria, G., Muthukumar, V., Parvania, M., Peisert, S., McParl, C., & Scaglione, A. (2014). A hybrid network IDS for protective digital relays in the power transmission grid. In *2014 IEEE international conference on smart grid communications (SmartGridComm)* (pp. 908–913). http://dx.doi.org/10.1109/SmartGridComm.2014.7007764.

Lee, J. M., & Hong, S. (2020). Keeping host sanity for security of the SCADA systems. *IEEE Access*, *8*, 62954–62968. http://dx.doi.org/10.1109/ACCESS.2020.2983179.

Li, D., Guo, H., Zhou, J., Zhou, L., & Wong, J. W. (2019). SCADAWall: A CPI-enabled firewall model for SCADA security. *Computers & Security*, [ISSN: 0167-4048] *80*, 134–154.

Li, X., & Hedman, K. W. (2020). Enhancing power system cyber-security with systematic two-stage detection strategy. *IEEE Transactions on Power Systems*, *35*(2), 1549–1561. http://dx.doi.org/10.1109/TPWRS.2019.2942333.

Liang, H., Ye, C., Zhou, Y., & Yang, H. (2021). Anomaly detection based on edge computing framework for AMI. In *2021 IEEE international conference on electrical engineering and mechatronics technology* (pp. 385–390). http://dx.doi.org/10.1109/ICEEMT52412.2021.9601888.

Lin, W. H., Lin, H. C., Wang, P., Wu, B. H., & Tsai, J. Y. (2018). Using convolutional neural networks to network intrusion detection for cyber threats. In *2018 IEEE international conference on applied system invention* (pp. 1107–1110). http://dx.doi.org/10.1109/ICASI.2018.8394474.

Liu, X., Li, Z., Shuai, Z., & Wen, Y. (2017). Cyber attacks against the economic operation of power systems: A fast solution. *IEEE Transactions on Smart Grid*, *8*(2), 1023–1025. http://dx.doi.org/10.1109/TSG.2016.2623983.

Lopez Perez, R., Adamsky, F., Soua, R., & Engel, T. (2018). Machine learning for reliable network attack detection in SCADA systems. In *2018 17th IEEE international conference on trust, security and privacy in computing and communications/ 12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)* (pp. 633–638). http://dx.doi.org/10.1109/TrustCom/BigDataSE.2018.00094.A.

Mahmud, R., Vallakati, R., Mukherjee, A., Ranganathan, P., & Nejadpak, A. (2015). A survey on smart grid metering infrastructures: Threats and solutions. In *2015 IEEE international conference on electro/information technology* (pp. 386–391). http://dx.doi.org/10.1109/EIT.2015.7293374.

Mohan, S. N., Ravikumar, G., & Govindarasu, M. (2020). Distributed intrusion detection system using semantic-based rules for SCADA in smart grid. In *2020 IEEE/PES transmission and distribution conference and exposition (T & D)* (pp. 1–5). http://dx.doi.org/10.1109/TD39804.2020.9299960.

Oyewole, P. A., & Jayaweera, D. (2020). Power system security with cyber-physical power system operation. *IEEE Access*, *8*, 179970–179982. http://dx.doi.org/10.1109/ACCESS.2020.3028222.

Peng, Y. (2020). Application of convolutional neural network in intrusion detection. In *2020 international conference on advance in ambient computing and intelligence* (pp. 169–172). http://dx.doi.org/10.1109/ICAACI50733.2020.00043.

Peng, W., Kong, X., Peng, G., Li, X., & Wang, Z. (2019). Network intrusion detection based on deep learning. In *2019 international conference on communications, information system and computer engineering* (pp. 431–435). http://dx.doi.org/10.1109/CISCE.2019.00102.

Radoglou-Grammatikis, P. I., & Sarigiannidis, P. G. (2018). An anomaly-based intrusion detection system for the smart grid based on CART decision tree. In *2018 global information infrastructure and networking symposium* (pp. 1–5). http://dx.doi.org/10.1109/GIIS.2018.8635743.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*.

Sharafaldin, I., Habibi, A. L., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*.

Shum, J., & Malki, H. A. (2008). Network intrusion detection system using neural networks. In *2008 fourth international conference on natural computation* (pp. 242–246). http://dx.doi.org/10.1109/ICNC.2008.900.

Singh, V. K., Ebrahem, H., & Govindarasu, M. (2018). Security evaluation of two intrusion detection systems in smart grid SCADA environment. In *2018 north American power symposium* (pp. 1–6). http://dx.doi.org/10.1109/NAPS.2018.8600548.

Siniosoglou, I., Radoglou-Grammatikis, P., Efstathopoulos, G., Fouliras, P., & Sarigiannidis, P. (2021). A unified deep learning anomaly detection and classification approach for smart grid environments. *IEEE Transactions on Network and Service Management*, *18*(2), 1137–1151. http://dx.doi.org/10.1109/TNSM.2021.3078381.

Subasi, A., et al. (2018). Intrusion detection in smart grid using data mining techniques. In *2018 21st Saudi computer society national computer conference* (pp. 1–6). http://dx.doi.org/10.1109/NCG.2018.8593124.

Sun, Y., Guan, X., Liu, T., & Liu, Y. (2013). A cyber–physical monitoring system for attack detection in smart grid. In *2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS)* (pp. 33–34). http://dx.doi.org/10.1109/INFCOMW.2013.6970712.

Suppitaksakul, C., & Saelee, V. (2009). Application of artificial neural networks for electrical losses estimation in three-phase transformer. In *2009 6th international conference on electrical engineering/electronics, computer, telecommunications and information technology* (pp. 248–251). http://dx.doi.org/10.1109/ECTICON.2009.5137002.

Talha, B., & Ray, A. (2016). A framework for MAC layer wireless intrusion detection & response for smart grid applications. In *2016 IEEE 14th international conference on industrial informatics* (pp. 598–605). http://dx.doi.org/10.1109/INDIN.2016.7819232.

Ullah, I., & Mahmoud, Q. H. (2017). An intrusion detection framework for the smart grid. In *2017 IEEE 30th Canadian conference on electrical and computer engineering* (pp. 1–5). http://dx.doi.org/10.1109/CCECE.2017.7946654.

Valdes, A., Macwan, R., & Backes, M. (2016). Anomaly detection in electrical substation circuits via unsupervised machine learning. In *2016 IEEE 17th international conference on information reuse and integration* (pp. 500–505). http://dx.doi.org/10.1109/IRI.2016.74.

Vijayanand, R., Devaraj, D., & Kannapiran, B. (2019). A novel deep learning based intrusion detection system for smart meter communication network. In *2019 IEEE international conference on intelligent techniques in control, optimization and signal processing* (pp. 1–3). http://dx.doi.org/10.1109/INCOS45849.2019.8951344.

Xu, Y. (2020). A review of cyber security risks of power systems: from static to dynamic false data attacks. *Protection and Control of Modern Power Systems*, *5*, 19. http://dx.doi.org/10.1186/s41601-020-00164-w.

Yang, L., Liu, J., & Zhang, Y. (2019). An intelligent security defensive model of SCADA based on multi-agent in oil and gas fields. *International Journal of Pattern Recognition and Artificial Intelligence*, *34*, http://dx.doi.org/10.1142/S021800142059003X.

Yang, Y., McLaughlin, K., Littler, T., Sezer, S., Pranggono, B., & Wang, H. F. (2013). Intrusion detection system for IEC 60870 − 5 − 104 based SCADA networks. In *2013 IEEE power & energy society general meeting* (pp. 1–5). http://dx.doi.org/10.1109/PESMG.2013.6672100.

Yohanandhan, R. V., Elavarasan, R. M., Manoharan, P., & Mihet-Popa, L. (2020). Cyber-physical power system (CPPS): A review on modeling, simulation, and analysis with cyber security applications. *IEEE Access*, *8*, 151019–151064. http://dx.doi.org/10.1109/ACCESS.2020.3016826.

Zhao, J., Chen, M., & Luo, Q. (2011). Research of intrusion detection system based on neural networks. In *2011 IEEE 3rd international conference on communication software and networks* (pp. 174–178). http://dx.doi.org/10.1109/ICCSN.2011.6013688.

Publication III

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 January 2023, accepted 14 February 2023, date of publication 22 February 2023, date of current version 27 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3247193

RESEARCH ARTICLE

Cyber Security in Power Systems Using Meta-Heuristic and Deep Learning Algorithms

SAYAWU YAKUBU DIABA, (Graduate Student Member, IEEE), MIADREZA SHAFIE-KHAH, (Senior Member, IEEE), AND MOHAMMED ELMUSRATI, (Senior Member, IEEE)
School of Technology and Innovations, University of Vaasa, 65200 Vaasa, Finland
Corresponding author: Sayawu Yakubu Diaba (sdiaba@uwasa.fi)

ABSTRACT Supervisory Control and Data Acquisition system linked to Intelligent Electronic Devices over a communication network keeps an eye on smart grids' performance and safety. The lack of algorithms protecting the power system communication protocols makes them vulnerable to cyberattacks, which can result in a hacker introducing false data into the operational network. This can result in delayed attack detection, which might harm the infrastructure, cause financial loss, or even result in fatalities. Similarly, attackers may be able to feed the system with fake information to hoax the operator and the algorithm into making bad decisions at crucial moments. This paper attempts to identify and classify such cyber-attacks by using numerous deep learning algorithms and optimizing the data features with a metaheuristic algorithm. We proposed a Restricted Boltzmann Machine-based nature-inspired artificial root foraging optimization algorithm. Using a publicly available dataset produced in Mississippi State University's Oak Ridge National Laboratory, simulations are run on the Jupiter Notebook. Traditional supervised machine learning algorithms like Artificial Neural Networks, Convolutional Neural Networks, and Support Vector Machines are measured with the proposed algorithm to demonstrate the effectiveness of the algorithms. Simulations show that the proposed algorithm produced superior results, with an accuracy of 97.8% for binary classification, 95.6% for three-class classification, and 94.3% for multi-class classification. Thereby outperforming its counterpart algorithms in terms of accuracy, precision, recall, and f1 score.

INDEX TERMS Artificial neural network, artificial root foraging, cyber security, deep learning, machine learning, metaheuristic algorithm, restricted Boltzmann machines, supervisory control and data acquisition, smart grid.

I. INTRODUCTION

The extraordinarily intricate architectural design of the electrical power systems must be handled cautiously and with the best control strategy feasible to ensure both the protection of human life and the system's safety [1]. The system becomes more complex as the control process must run more quickly [2]. Automated devices are introduced to modern power systems to make operating them easier. The number of pieces of protective equipment that are part of the system is directly impacted by operational demand and consumer count [3]. Recent years have seen the development of automated systems for connected power module protection, automation, and control [4]. Protective device performances have somewhat improved as a result of developments in algorithms and power systems architecture [5], [6].

However, the likelihood of security problems increases as the number of connections to the power system modules intensifies. Hence the quality of control is expected to be in the higher range for modern power systems. The contemporary power systems are implemented with various International Electrotechnical Commission (IEC) standards [7], [8] and are generally operated with six significant components, as depicted in Figure 1. Generators, transformers, and safety

The associate editor coordinating the review of this manuscript and approving it for publication was Christos Anagnostopoulos.

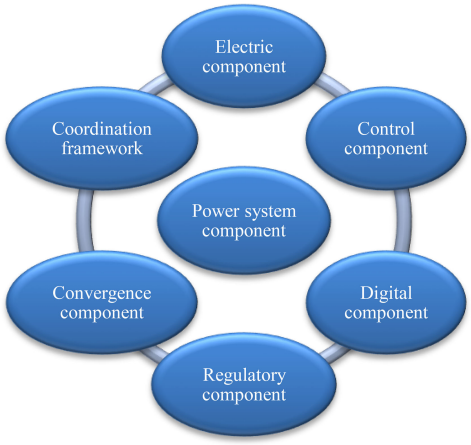


FIGURE 1. Component of the power system.

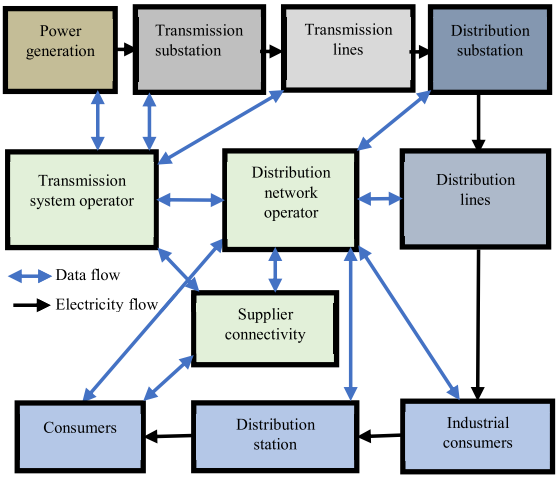


FIGURE 2. The architecture of a smart grid system.

equipment are all part of the power system's electrical components. These primary hardware ranges and ratings change depending on the loads connected to the network. The protection mechanisms built into the electrical system also differ depending on the linked equipment's location and nature [9]. The control components include the synchronization model and operational modules for transmitting the required signal to the digital modules used for the operation. The power system's information and communication devices, which transmit control signals between linked systems and components across wired or wireless networks, are represented by digital modules [10]. The convergence network regulates the power flow in the connected system by analyzing the load requirement and the power system state. The importance of the convergence networks increases when the power system is linked to Distributed Energy Resources (DERs) [11], [12]. The regulatory components ensure that the integration of power is constantly smooth and efficient.

In order to solve the problems with conventional digital components, which were designed to have certain limitations, smart grid power systems were developed. This is achieved by integrating distributed intelligence algorithms into the system. The distributed intelligence algorithms swiftly and efficiently support making decisions on the present digital components [13].

Smart grids, however, have more security concerns due to the distributed location of the control units. The architecture of the smart grid power systems includes the following four layers [14]. *Physical Layer:* It is identical to the layer found in every fundamental power system, which consists of a generation station, transmission lines, and a distribution unit. *Communication Layer:* The layer between the user and the service provider; this layer offers a network that allows for the discovery of the status of the power system's operation. *System Integration Layer:* This layer includes the computing and security infrastructure. It controls the data analytics process so that the control units can make several decisions. This is realized by importing a powerful algorithmic model. *Software Layer:* It enables the service provider to access the power consumption details from the user side. This layer provides information about the user and their nature to the system integration layer for future predictions.

Based on their general characteristics, the four kinds of cyber security problems for smart grid systems may be classified. They are issues with, *connectivity*, *trust*, *privacy*, and *software vulnerability* [15], [16].

Connectivity: Compared to other physical systems, the systems that make up the smart grid are more widely distributed. As a result, the smart grid power system's communication protocol necessitates constant operation and higher data transmission rates. The system transfers the data regularly; it poses numerous security concerns for the models. *Trust:* The smart grid systems are open to everyone. Some key equipment, specifically the Automated Meter Infrastructure (AMI) is situated in the user area. As a result, there is a greater chance that the system may be interfered with, and this risk is directly correlated with the user's level of trust, given that operational costs and other factors are involved. *Privacy:* The smart meters connected to the system contain the user's basic information, which is the most targeted device for intruders. *Software Vulnerabilities:* The smart grid systems are mostly monitored with Supervisory Control and Data Acquisition (SCADA) computer software. The SCADA system's modernization, standardization of communication protocols, and increasing interconnectivity have all contributed to a sharp rise in cyberattacks on the system over time, rendering it vulnerable to assault from anywhere around the globe [16]. Hence it is a must to protect the smart grids' SCADA systems from malicious cyber-attacks and malware disruption.

The aforementioned issues prompted the following goals for this study, which are as follows:

1) To employ a nature-inspired artificial root foraging optimization algorithm with a Restricted Boltzmann

Machine (RBM), to provide an enhanced algorithm that reliably detects and classifies attack intrusions in the smart grids' SCADA systems.
2) Enhanced adaptability: Nature-inspired optimization algorithms are designed to be highly adaptable, and can be modified or fine-tuned to meet the specific needs of a particular application. By combining an RBM with a nature-inspired algorithm, we seek to create a system that is highly adaptable and able to learn and adapt to new threats as they arise.
3) Increased efficiency: Nature-inspired optimization algorithms are typically more efficient than traditional optimization methods, as they are able to explore a more extensive search space more quickly. By combining an RBM with a nature-inspired algorithm, we seek to propose an algorithm that is able to analyze large amounts of data more quickly and efficiently, allowing for faster and more effective threat detection.
4) Reduced reliance on labelled data: RBMs are capable of performing unsupervised learning, which means they can learn from data that is not labelled or categorized. By combining an RBM with a nature-inspired algorithm, it is possible to create a system that can learn from a larger and more diverse dataset, which may be particularly useful in cases where labelled data is scarce or difficult to obtain.
5) To demonstrate the performance of the proposed algorithm's efficiency to other existing algorithms in terms of accuracy, precision, recall, and f1 score.

Section I captures the introduction of the paper. Section II contains background information on related studies and theoretical frameworks from the literature. The proposed algorithm is covered in Section III, while the simulation results are described in Section IV. Section V serves as the paper's conclusion.

II. RELATED STUDIES

The smart grid protection strategy uses local measures or external devices to build a smart grid protection system that is both effective and efficient. However, one of the key issues is the ability to connect physical and digital components to suit the configuration of the system. Measurement of data source authentication system was developed to analyze the data flow of a power system by extracting the features through an ensemble empirical mode decomposition model with the Fast Fourier Transform (FFT) technique. The experiment was conducted with a back-propagation neural network for data classification. An accuracy of 80.9% is achieved, and comparatively, it is better than the traditional long short-term memory (LSTM) model's accuracy of 77.8% [17]. To train the neural network algorithms, a sizable dataset is required. The performance of a neural network algorithm's prediction process is influenced by the amount of training data present in the network. The authors of [18] generated a power system dataset based on IEC 61850 Generic Object-Oriented Substation Event (GOOSE) communication for developing a reliable cybersecurity system.

The components of the power system are divided into numerous categories to monitor the load demand in different areas. Due to environmental conditions, the associated field will see variations in demand in particular. The system is more vulnerable to cyber threats since the scattered devices are connected through different channels [19]. The testbed-based power system quality analysis is one of the familiar methods widely used for observing the response of the power system in different scenarios. The test bed generates different kinds of cyber security issues to analyze and formulate a defending algorithm. An OMNeT++-based simulation technique was structured [20] to analyze the nature of cyberattacks in a bidirectional communication network. The model was integrated with Power Systems Computer Aided Design (PSCAD) for the power simulation.

The physical power systems are open to dynamic data injection attacks. An example is the ease with which the energy consumption values on smart meters could be altered. So, an interval state estimation method was developed to analyze the possible variations in the readings with respect to time. A kernel quantile regression is also incorporated in the work to estimate the uncertainties in renewable and electric load forecasting applications [21]. The cyberattack on the Internet of Things (IoT)-based smart grids may affect the costly and important systems that are connected to the power system. The hospital equipment and electric train are some of the costlier and most needed systems that always depend upon the quality of the power supply. Therefore, a blockchain-based technique was equipped with Hilbert-Huang transform to estimate power quality through the data collected from voltage and current sensors. The experimental work founds satisfied with the performance of the proposed model on false data injection attacks [22].

The false data injection process can also be observed by estimating the phasor measurements of the connected loads. A two-layer defense system was developed [23] to observe the change in the values of the power system. The defense resources are optimized in the work with a zero-sum static game algorithm. It is demonstrated that the proposed two-layer model is useful for examining false data injection attacks. Providing cybersecurity to DER, such as photovoltaic systems (PV), is one of the challenging tasks in power systems. To accomplish this, the connected syste's active and reactive power is analyzed along with its permitted voltage level for transmission. The syste's network topology is used to observe the power changes on each terminal. The change in the difference in various estimations makes the work to predict the attack output on its class [24]. A decision-making algorithm was outlined to estimate the cyberattacks in multi-microgrid systems. A fuzzy static Bayesian game model was utilized in the work for predicting the optimal security strategy, and a hybrid approach based on a fuzzy algorithm was used to reach a consensus [25].

A cybersecurity risk management system was developed to predict attacks in cyber-physical systems. The work analyzes the criticality of the assets in cyberattacks and their effect on the output of the system. The attack scenario, control, and threats are considered in the work for estimations [26]. A stochastic coupling strategy was designed to estimate the cascading process in cyber-physical systems. This has been performed by keeping two asymmetric subnetworks for increasing the accuracy of random and frequent cyberattacks. The experimental projection indicates a reduced estimation time for frequent attacks over the random models [27]. A deep reinforcement learning technique was structured to provide cybersecurity protection on distributed power systems. The performance of the system was experimented on the IEEE 13-bus model and the simulation results are not found satisfactory under the greedy attack conditions [28].

When responding to hostile attacks on industrial control systems, machine learning techniques are particularly accustomed. The results of an experiment using the random forest and J48 algorithms to identify intrusions in control systems were found to be good in forecasting cyber-attack behaviors [29]. A dimensionality reduction and statistical hypothesis techniques were merged to ensure cybersecurity on smart grids. A concept drift methodology was utilized in the work to observe the differences between the physical grid change and data manipulation. Experimental work was performed in the work with and without concept drift and found satisfactory with the concept drift technique [30]. A physics-informed spline learning technique was developed to detect anomalies in power electronic circuits. The experiment was found satisfactory even when trained with minimal data [31].

The review of the literature looks at the various strategies developed to address security issues in power systems. The majority of the systems, however, were created to recognize the introduction of false data into power systems. This was accomplished by analyzing the system's typical behavior to anticipate the system's abnormal response when fictitious data was injected. Because their analysis is feature-based, deep learning and machine learning algorithms are quite good at making these kinds of predictions. In the part that follows, a feature optimization technique based on a meta-heuristics algorithm is used to assess the effectiveness of deep learning-based algorithms to observe security vulnerabilities in SCADA systems for smart grids.

III. METHODOLOGY

The overall artificial root foraging, RMB architecture, and our variation are all introduced in this section.

A. OVERVIEW OF THE SYSTEM

The proposed model utilizes a nature-inspired artificial root foraging method for optimizing the information collected through the power systems sensor and data transmitters. Voltage and power sensors are used to detect the anomaly of the power system; the abnormality of the power system is observed and forwarded to the base station through an IoT network. The receiving station tabulates the collected information and projects the outcome as a database. The dataset creation process makes the base station verify all the collected information and separates the readings that came up with errors and missing information. The dataset creation process can be limited with respect to time as it may provide the amount of data to be stored in the database. Figure 3 represents the workflow of the proposed model.

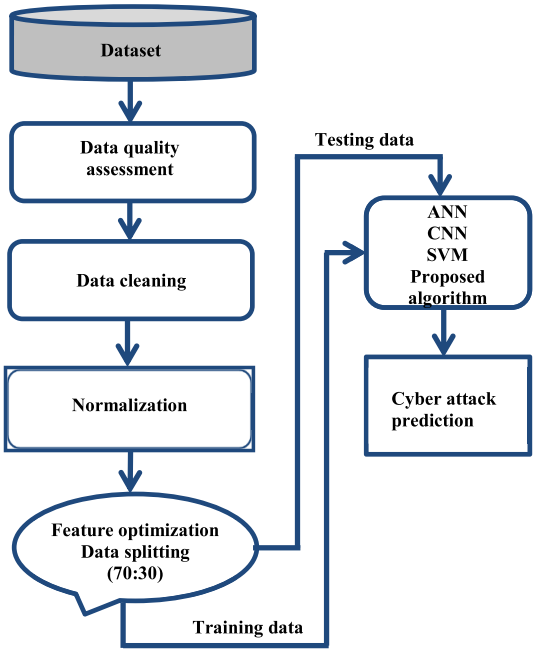


FIGURE 3. The workflow of the proposed model.

B. PREPROCESSING

Preprocessing is the fundamental technique for organizing the data gathered from the remote terminal unit (RTU) and other Intelligent Electronic Devices (IED) modules. In this step, the unstructured and unformatted data are organized to make the information reliable before it is used in the training process.

In general, the data can be segregated into two categories: numerical data and categorical data. The binary format is used for categorical data, and whole numbers or fractions are used for numerical data. Information about the power system is gathered in numerical form for the proposed task.

The quality of the feature extraction mostly depends on the caliber of the data used in the operation. Therefore, the paper makes use of the data translation process, data cleaning process, and data quality assessment process. As previously shown, the data quality assessment sends the available data to the data cleaning process while moving the missing data to the trash. The data cleaning procedure enables the removal of duplicate data and requires the manual insertion of data when it is discovered to be abnormal or missing.

C. ARTIFICIAL ROOT FORAGING OPTIMIZATION

1) CLASSICAL PLANT ROOT GROWTH MODEL

The biological root growth optimization algorithm served as the basis for designing the artificial root foraging optimization algorithm. A biological plant's primary root advances toward the ground, while its lateral roots spread outward like a branch from the main root. Similarly, the lateral roots are also permitted to develop numerous lateral roots in diverse directions. While the primary roots are not permitted to do so, the lateral roots are permitted to form in all directions with varying degrees of movement. Hence, the artificial root foraging algorithm is also constructed using the conventional optimization model that is used to predict the growth of plant roots. Root growth is thought to be hindered by the nature of the soil, and the main root movement and lateral root movement are thought to be the best solutions. The change in direction and length adjustments are regarded as the fine-tuning parameters for the problems [32]. The following factors are considered for ideal plant growth, and the same has been followed in the artificial model.

Factor 1: The spatial structure of the roots is heavily influenced by the auxin concentration in the plants. It allows the root to be automatically structured by observing the problem.

Factor 2: A single root apex advances in the same direction and can generate children's root apices.

Factor 3: Auxin availability causes the root system to develop a variety of lateral roots and branches.

Factor 4: Hydrotropism allows the tip of the main root and lateral roots to move in their respective directions along the trajectory.

2) AUXIN REGULATION

The auxin concentration is the primary parameter for developing a new branch count and movement operations [33]. Therefore, the nutrition availability of the soil is formulated as follows.

$$f_x = \frac{fitness_x - f_{low}}{f_{high} - f_{low}} \tag{1}$$

Mathematically, the auxin concentration is written as

$$A_x = \frac{f_x}{\sum_{y=1}^{s} f_x} \tag{2}$$

where the function value is $fines s_x$, f_x is the normalization value of the root fitness, f_{high} and f_{low} represent the current root population count and s is the population size.

3) STRATEGY ON MAIN ROOT GROWTH

The growing probability of the main root is free from the probability of branch and re-growing factor. The movement of the main root depends upon the best individual operation formulated from its current position [34]. It is mathematically represented as

$$I_x^t = I_x^{t-1} + l.\varepsilon.\left(I_{lbest} - I_x^{t-1}\right) \tag{3}$$

here, I_x^t implies a new location, I_x^{t-1} represents the location of root x. Learning inertia takes l, ε is the uniform random coefficient between 0 and 1 and I_{lbest} stands for the best individual from the present location.

4) BRANCHING OPERATOR

The branching operator develops a new individual based on the root apex estimations. It is predicted by estimating the available auxin concentration over the threshold value included in the branch [35]. The number of individuals generated from the branch is calculated as

$$\begin{cases} branch\ individuals\ \mathrm{w}_x & \text{if } A_x > threshold\ value \\ stop\ branching & otherwise \end{cases} \tag{4}$$

Therefore, the numbers of newly generated apices are estimated from the following equatio

$$W_x = \varepsilon.A_x(B_{\max} - B_{\min}) + B_{\min} \tag{5}$$

ε is the uniform random coefficient between 0 and 1, A_x is the auxin concentration level at the root. B_{max} and B_{min} represent the branching count. The location for developing a new branch root is predicted from the primary root through Gaussian distribution $N(I_x^t, \sigma^2)$. The standard deviation is written as

$$\sigma = \left(\frac{x_{\max} - x}{x_{\max}}\right)^2 \times \left(\sigma_{ini} - \sigma_{fin}\right) + \sigma_{fin} \tag{6}$$

where x_{max} is the maximum iteration, i is the current iteration index, σ_{ini} is the initial standard deviation, and σ_{fin} is the final standard deviation.

5) LATERAL OR BRANCH ROOT GROWTH

The lateral roots are allowed to conduct a random search on every feeding state [36], [37]. The length and growing degree of the lateral roots are changed between each other, and that can be mathematically projected as

$$I_x^t = I_x^{t-1} + \varepsilon\ (l_{\max} D_i * \phi) \tag{7}$$

$$\phi = \frac{\delta_i}{\sqrt{\delta_i^T \times \delta_i}} \tag{8}$$

where l_{max} stands for the maximum length of the lateral root, D_i is the dimension growth direction of the lateral root i, and ϕ stands for the growth angle formulated with a random vector δ_i.

6) DEAD ROOT GROWTH SHRINKABLE

The growing process might not be supported by the roots if they were unable to absorb nutrients. The auxin distribution evaluates the likelihood that the lateral roots will grow and, if they do not, they are removed from the main root.

D. RESTRICTED BOLTZMANN MACHINES

The RBM technique was primarily created for regression, feature learning, and dimensionality reduction applications. It is a subset of the family of energy-based models, where

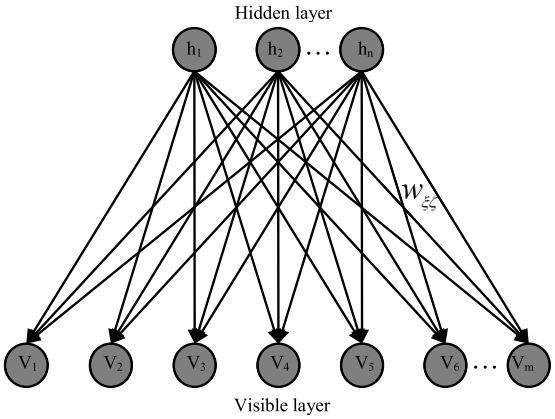


FIGURE 4. The architecture of the RBM.

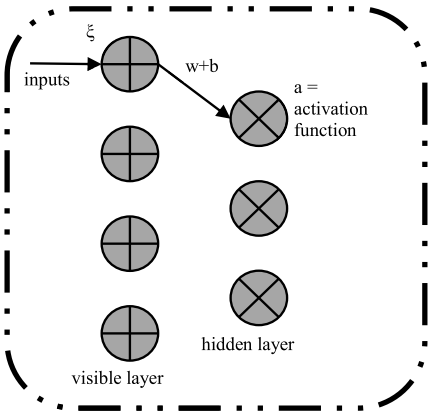


FIGURE 5. The input path of an RBM.

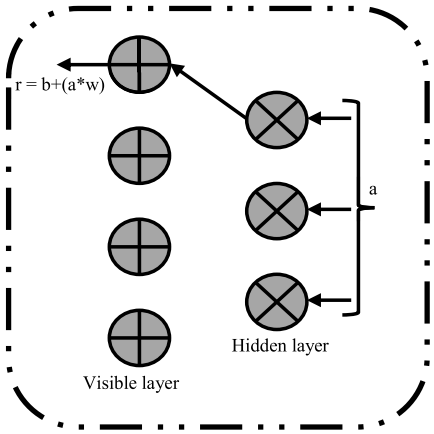


FIGURE 6. The reconstruction of RBM.

each configuration of the relevant variables corresponds to a training-relevant finite scalar energy value. The RBM algorithms are typically shallow and only use two levels of network connection [38]. As a result of their simplicity, RBMs are widely used in a variety of applications. The primary layer of the RBM is represented as the visible layer, and the second layer is mentioned as the hidden layer. The number of neural nodes included in the layer varies with respect to the count of inputs made to the approach and the interconnection between the nodes makes a neurological connection like a human brain. The RBM connections are very special, and there the intra-connections are restricted. The node analyzes the input received by the, computes it, and decides whether to permit it or not for neighbor node connection [39]. The bipartite interactional graph of the RBM is depicted in Figure 4.

The feature that the visible layer node collects is denoted by the letter ξ and it is passed to the hidden layer by multiplying the weighted value w and adding the bias b [40]. The following expression can be used to describe the outcome of this operation as an activation function of the supplied input.

$$f((\xi \times w) + b) = a \tag{9}$$

where f represents the activation function, ξ is the input, and w stands for the weights. The bias is represented by b and a is for the activation function.

The hidden layer activations are considered as input in the reconstruction step, where the input is given to the hidden layer. Same as the input path, the reconstruction model also operates the input with the same multiplication factor. Hence the output gives a value to the original input. Figures 5 and 6 indicate the input path of an RBM and the reconstruction model of the RBM, respectively.

Generally, the values of the weights included are assumed randomly, and presumably, there will always be a huge deviation between the input and output of the RBM. So, the weights are modified continuously to reduce the error observations in estimating the reconstruction r value. The nodes are designed to take the low-level feature present in all the attributes available in the dataset. This paper considers that the RBM has a total of n visible neurons as $v = v_1, v_2, \ldots, v_n$ and total hidden neurons m has hidden neurons as $h = h_1, h_2, \ldots, h_m$. The model uses binary values since the study examines the binary problem (natural or attack) of the existence of anomalies. the random variable takes the values (v, h) $\in$. $\{0, 1\}^{m+n}$. Thus, the probability distribution according to [41] can be written as

$$P(v, h) = \frac{1}{z} e^{-E(v,h)} \tag{10}$$

Z is the partition function. An energy function E(v,h) of the model can be defined as [42] and [43]

$$E(v, h) = -\sum_{\xi=1}^{n}\sum_{\zeta=1}^{m} w_{\xi\zeta} h_m v_n - \sum_{\xi=1}^{n} g_n v_n - \sum_{\xi=1}^{m} q_m h_m \tag{11}$$

TABLE 1. Description of the employed dataset.

Data class	Data details	Event count out
Binary classification	Natural event	9
	Attack event	28
Three-Classification	No event	1
	Natural event	8
Multiclass classification	Attack event	28
	All class	37

Equation (11) can be re-written as

$$E(\mathbf{v}, \mathbf{h}) = \mathbf{g}^T \mathrm{v} - \mathrm{q}^T \mathbf{h} - \mathbf{v}^T \mathbf{W}\mathbf{h} \tag{12}$$

where the considered features for the training process of ξ is $\xi \in \{1, 2, \ldots, \mathrm{n}\}$ and $\zeta \in \{1, 2, \ldots, m\}$. The weight is denoted by $\mathrm{w}_{\xi\zeta}$, g_n is the n^{th} feature of the ξ^{th} input of the v^{th} visible neurons. Similarly, q_m is the m^{th} feature of the ζ^{th} input of the h^{th} hidden neuron. Due to the RBM's bipartite nature, there is no connection between a hidden neuron and a hidden neuron, just as there is no connection between a visible neuron and a visible neuron. The model for conditional independence is described as

$$p(\mathbf{v}, \mathbf{h}) = \Pi_{\zeta=1}^{m} p(v_\zeta | \mathbf{h}) \tag{13}$$

$$p(\mathbf{v}, \mathbf{h}) = \Pi_{\xi=1}^{n} p(\mathbf{v}_\xi | \mathbf{h}) \tag{14}$$

E. DATA DESCRIPTION

This paper utilizes the power system attack detection dataset developed by the Oak Ridge national laboratory of Mississippi State University [44]. The dataset is separated into three types, binary class, three class, and multi-class. It is created from a single dataset consisting of 15 sets of information from 37 types of power system events. Except for the multi-class dataset, the details are in CSV format. The content of the dataset is shown in Table 1.

Figure 7 shows a three-bus two-line transmission system modified from the IEEE four-bus three-generator system, it explores the architectural view of the test framework used for the analysis. Despite being a very modest system, it embodies the core of the broader power system and is simple enough to be understood in its entirety. The classifier suggested in this work would be used multiple times to monitor different parts of a power system. The framework merges two generator models consisting of four IEDs, specifically, relays (R_1 to R_4) for providing a switching operation to the circuit breakers (Bk_1 to Bk_4). Each circuit breaker is connected with a separate IED [44]. Therefore, it trips off the breaker unit when a real or fake fault is detected in the circuit. The IEDs are not equipped with any algorithm so far for analyzing the nature of the fault. Thus, this kind of model requires a manual operation to re-enable the circuit from its faulty condition. The major type of faults and attacks that can happen in a power system model is as follows [45].

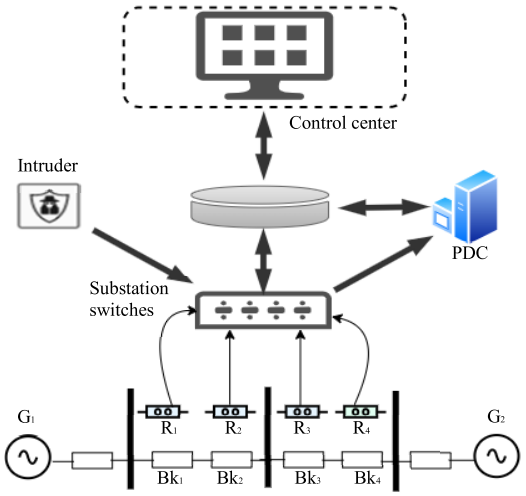


FIGURE 7. Overview of the power system framework.

1) FAULTS

a: SHORT CIRCUIT

These kinds of faults may happen in a power system owing to natural and manual errors at any location. The location of the fault can be identified by observing the current and voltage changes in the circuit. A short circuit fault in a power system occurs when there is an abnormal connection between two points in the electrical circuit that are not intended to be connected. This can cause a sudden and large increase in the flow of electrical current, which can damage or destroy electrical equipment and pose a risk of injury to personnel. Short circuit faults can be caused by a variety of factors, including damaged or faulty electrical components, loose connections, and the presence of foreign objects or debris in the electrical circuit. They can also be caused by natural disasters such as lightning strikes or earthquakes [46].

When a short circuit fault occurs, the electrical system is designed to automatically detect the fault and interrupt the flow of current to prevent damage to the equipment and protect personnel. This is typically done by using protective devices such as circuit breakers, fuses, and relays, which are designed to detect abnormal electrical conditions and interrupt the flow of current. It is important to promptly address short circuit faults in order to minimize the risk of damage to the electrical system and ensure the safe and reliable operation of the power system. This may involve identifying and repairing the root cause of the fault, as well as testing and inspecting the affected equipment to ensure it is safe to return to service [46].

b: LINE MAINTENANCE

For the duration of the maintenance period, the relay modules connected to the power system model are disconnected from the circuit. These kinds of errors are intentional and are

simple to fix. Line maintenance in power systems refers to the activities that are performed to ensure that transmission and distribution lines are operating safely and efficiently. These activities can include inspections, repairs, and upgrades of transmission and distribution lines, as well as the associated equipment such as transformers, switches, and other electrical components.

Line maintenance is an essential part of the overall operation and maintenance of a power system, as it helps to ensure the reliability and safety of the electrical grid. Line maintenance activities can be performed on both overhead and underground transmission and distribution lines, and may involve a range of tasks, such as: Inspecting and testing electrical equipment to identify any potential issues or problems. Replacing damaged or worn-out components. Upgrading equipment to improve performance or increase capacity. Cleaning and maintaining transmission and distribution lines to remove debris and vegetation that could cause problems. Performing preventive maintenance activities to prevent potential problems from occurring. Line maintenance is typically carried out by trained and certified professionals with the necessary knowledge and skills to work safely on high-voltage electrical equipment. In some cases, specialized equipment such as bucket trucks or aerial lifts may be used to access transmission and distribution lines for maintenance activities [46].

2) ATTACKS

a: DATA INJECTION ATTACK

A data injection attack in power systems, also known as a manipulation attack, is a type of cyber-attack that involves injecting false or malicious data into the control systems of a power grid. The goal of this type of attack is to disrupt the normal operation of the power grid and potentially cause damage to the system.

Data injection attacks can take many different forms, but they generally involve the attacker injecting false or malicious data into the control systems of the power grid to mislead the operators or cause the system to malfunction. For example, an attacker might inject false data into the control systems of a power grid to indicate that there is a fault in the system, when in fact there is not. This could lead to the operators taking inappropriate or unnecessary actions to respond to the false fault, which could potentially cause damage to the power grid.

Data injection attacks can be difficult to detect, as they often involve the injection of small amounts of false data into the control systems of the power grid. They can also be difficult to prevent, as they require a high level of access to the control systems of the power grid. Power grid operators need to implement robust cybersecurity measures to protect against these types of attacks [45].

b: RELAY SETTINGS CHANGE ATTACK

A relay settings change attack in power systems is a type of cyber-attack that involves altering the settings of protective relays in the power grid. Protective relays are electrical devices that are used to automatically detect and respond to abnormal conditions in the power grid, such as short circuits or over currents. They are an essential component of the power grid's protection system, as they help to ensure the stability and reliability of the grid [45].

In a relay settings change attack, an attacker may attempt to manipulate the settings of protective relays to disrupt the regular operation of the power grid. For example, the attacker may change the settings of the relays so that they do not respond to certain types of fault conditions, or so that they respond in a way that is not appropriate for the specific fault condition. This can lead to widespread power outages and other disruptions in the power grid [45].

Relay settings change attacks can be challenging to detect, as they often involve subtle changes to the settings of the protective relays. They can also be difficult to prevent, as they require a high level of access to the power grid's control systems. Power grid operators need to implement robust cybersecurity measures to protect against these types of attacks [45].

c: TRIPPING COMMAND INJECTION ATTACK

It is a command kind of attack that makes the relay open the circuit with a command received from a remote location. A tripping command injection attack in power systems is a type of cyber-attack that involves injecting false or malicious commands into the control systems of a power grid in order to disrupt the normal operation of the system. The goal of this type of attack is to cause equipment to trip or shut down, potentially leading to widespread power outages and other disruptions in the power grid [45].

In a tripping command injection attack, an attacker may inject false or malicious commands into the control systems of the power grid in an effort to cause equipment to trip or shut down. For example, the attacker might inject a command to trip a circuit breaker or shut down a generator. This could lead to widespread power outages and other disruptions in the power grid. Tripping command injection attacks can be challenging to detect, as they often involve the injection of small amounts of false or malicious data into the control systems of the power grid. They can also be difficult to prevent, as they require a high level of access to the control systems of the power grid. To defend against these kinds of attacks, power grid operators must install strong cybersecurity safeguards [45].

F. DEEP LEARNING PERFORMANCE EVALUATION METRICS

Deep learning is a type of machine learning that uses deep neural networks to learn and make predictions or decisions. The performance metrics used to evaluate the effectiveness of a deep learning model are similar to those used for other types of machine learning models. Because of the task under study and the kind of model being employed, we concentrate only on the four threshold parameters that the classification problem's performance metric is defined by

TABLE 2. Confusion matrix for a binary classifier.

	Actual true	Actual false
Predicted true	True positive	False positive
Predicted false	False positive	True Negative

Accuracy: This is a common metric for classification tasks, and it is defined as the number of correct predictions made by the model divided by the total number of predictions. Mathematically represented as [56]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

Precision: This metric is used to measure the precision of a classifier, and it is defined as the number of true positive predictions made by the model divided by the total number of positive predictions [56].

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

Recall: This metric is used to measure the recall of a classifier, and it is defined as the number of true positive predictions made by the model divided by the number of positive cases in the dataset [56].

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

F1 score: This is a metric that combines precision and recall, and it is defined as the harmonic mean of precision and recall [56].

$$F1score = \frac{2\,(Presicion \times Recall)}{Presicion + Recall} \tag{18}$$

Equations (15), (16), (17), and (18) are derived using the confusion matrix. The confusion matrix is a table that is used to evaluate the performance of a classifier, and it is often used in conjunction with various performance metrics to provide a more complete picture of the classifier's effectiveness.

IV. EXPERIMENTAL ANALYSIS

The experiment was performed in a Jupyter notebook on a 16GB RAM Intel 7 processor system. The proposed RF-RBM technique was tested against conventional CNN, ANN, and SVM algorithms because those were found to be successful models in several intrusion detection studies [37], [49]. In this, the SVM is a machine learning-based technique, whereas CNN and ANN are deep learning-based techniques. We utilize the hyperparameters given in Table 3 for the simulations, and we classify the network intrusion through different algorithms.

One of the most used neural network algorithms, CNN, can provide a higher accuracy rate when the training data samples are plentiful. However, because CNN learns characteristics from a large dataset, preprocessing of the training data is minimal. Three layers make up a conventional CNN: a convolution layer, a pooling layer, and a fully connected layer. The convolution layer is set up to separate the kernel's learnable parameters from the input data. The kernel clarifies to the layer the kind of information that is available [47] and [49]. Data is forwarded by the kernel to different neurons in the pooling layer, which lowers the spatial complexity of the retrieved information in the convolution layer. All of the CNN's neurons are interconnected in the fully connected layer with their biases toward comprehending the data that has been gathered [50].

TABLE 3. Hyperparameter settings.

Model parameter	Total
Visible node	128
Hidden neurons for CNN, ANN	2
Batch size	128
Epoch	1000
Activation functions	ReLU, Sigmoid
Learning rate	0.1

The ANN is one of the successful models that can mimic the nature of the human brain. All neurons are interconnected between them as different layers, just like in a human brain. The input, output, and hidden layers are the principal layers of an ANN, and the number of hidden layers can be increased depending on the demands of a situation. The hidden layer is used to extract different features and patterns from the input data, while the input layer is used to provide diverse information to the neural network design. Additionally, the hidden layer applies a bias value to the gathered characteristics to do an efficient calculation [48], [51].

The SVM is a supervised machine learning technique that handles the classification problem by drawing the best distinction between the various classes. The optimal boundary line can be determined by locating an extreme vector point in the available dimension space. SVMs are frequently used for binary classification and can be applied to multiple classifications by generating a non-linear function that generates new variables as the kernel [49], [52].

In machine learning, feature selection is a crucial operation [53]. We opted for our algorithm because the meta-heuristic nature-inspired algorithm can provide a strong foundation for identifying patterns and anomalies in the data, by using the input and output without needing gradient information [54]. The RBM can be used to learn and recognize more complex features that may be indicative of an intrusion. Together, these two approaches can provide a powerful tool for detecting and responding to threats in smart grid systems.

A. RESULTS

The 15 sets of information from 37 types of power system events were combined into a single dataset. For the experiments in this paper, 70% of the data is used for training, and 30% is used for testing. Using the hyperparameter settings in Table 3, the three distinct experiments are conducted.

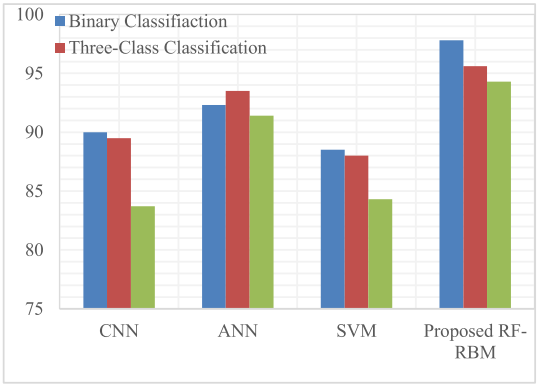


FIGURE 8. The accuracy of the conducted experiments.

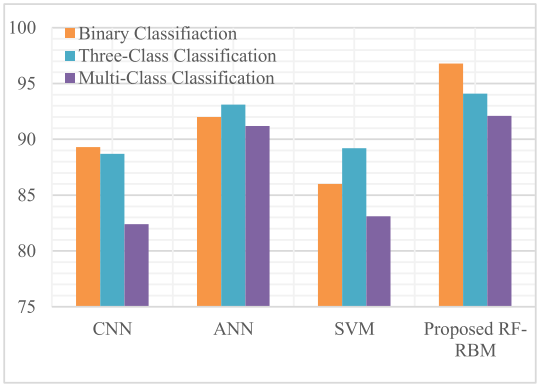


FIGURE 10. The recall score of the conducted experiments.

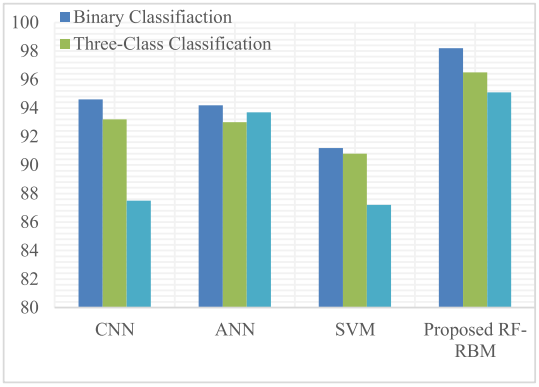


FIGURE 9. The precision of the conducted experiments.

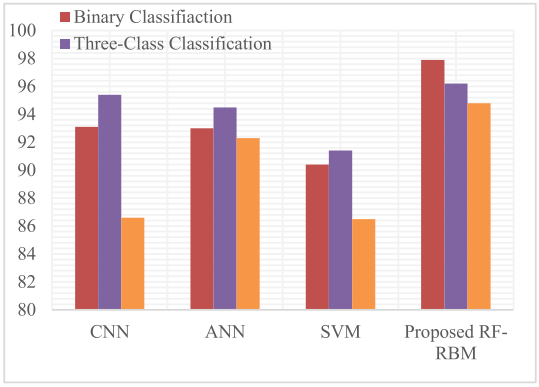


FIGURE 11. The f1 score of the conducted experiments.

Figures 8, 9, 10, and 11 show the experiment's findings, demonstrating the accuracy, precision, recall, and f1 score of the verified algorithms in more detail. Figure 8 depicts the performance of the verified algorithms measured in terms of accuracy across all three experiments. The results show that the accuracy of the algorithms in the binary classification experiment consistently outperformed the other two experiments, with the exception of the ANN algorithm in the three-class classification experiment. In this case, the ANN algorithm performed slightly better in the three-class classification experiment compared to the binary classification experiment and the multi-class classification.

According to the results depicted in Figure 9, the precision of the multi-class classification experiment improved considering the three-class classification experiment, but this improvement was only observed for the ANN algorithm. These results suggest that the ANN algorithm may be more effective at achieving higher precision in multi-class classification tasks. However, the performance of the multi-class classification experiment was subpar when utilizing the CNN and SVM algorithms.

As illustrated in Figure 10, the recall of the experiment revealed an improvement in the three-class classification for both the ANN and SVM algorithms compared to the other two experiments. The performance of the binary classification experiment is higher for the proposed RF-RBM because the sample counts on either one class in the binary classification are very large. However, the irregular distribution of the three-class classification experiment is a result of the significant drop in data for the no-event class, leading to a decrease in performance.

The results of the f1 score estimations shown in Figure 11 indicate that the outcomes of the three-class classification are better in all the experiments, except for the proposed RF-RBM. The proposed algorithm outperforms the other three algorithms in three-class classification and multi-class classification, but it extremely outperforms them in binary classification.

Furthermore, we compare the results of this paper to the result of comparable papers that employed the same dataset. The comparison using the binary classification dataset is shown in Table 4, and the three-class classification dataset is shown in Table 5.

TABLE 4. Comparison of models results with binary classification dataset.

Model	Accuracy	Precision	Recall	F1-score	Ref
SVM-AC	84.4	86	84.9	-	[52]
Linear SVM	76.2	76.2	75.4	73.3	[57]
GA-Linear SVM	87.0	85.2	86.6	80.1	[57]
RBF SVM	81	80.1	82.5	76	[57]
GA-RBF SVM	91.9	93.7	95	87	[57]
MLPNN	78.8	78.5	80.2	75.3	[57]
GA-MLPNN	86.4	87.2	85.7	84.9	[57]
RF	81.9	82.6	83.9	77.9	[57]
GA-RF	88.2	87.4	89.1	86.1	[57]
JRipper	-	85.0	70.0	-	[47]
PSO SVM	89.5	90.2	80.7	-	[51]
AdaBoost + JRipper	-	94	89	-	[47]
Proposed RF-RBM	**97.8**	**98.2**	**96.8**	**97.9**	

TABLE 5. Comparison of models results with three-class classification dataset.

Model	Accuracy	Precision	Recall	F1 score	Ref
SVM-ACO	78	80.5	77.4	NA	[56]
GA-RBF SVM	90.9	89.9	91.3	85.8	[57]
PSO-SVM	85.7	86.5	83.1	NA	[58]
AdaBoost + JRipper	99	95	**100**	NA	[47]
Proposed RF - RBM	**94.3**	**95.1**	92.1	**90.3**	

V. CONCLUSION

In this study, we present a nature-inspired restricted Boltzmann machine algorithm to detect and classify the types of attacks in the smart grids' SCADA systems. The fundamental notion is that the artificial root foraging optimization method is designed on the biological root growth optimization algorithm. To demonstrate the optimization capability, the dataset features were fine-tuned using the artificial root foraging algorithm before the neural network algorithm. The proposed RF-RBM algorithm is compared to three cutting-edge neural network algorithms in the experimental study, which was conducted in three categories: binary classification, three-class classification, and multi-class classification. The outcomes of the experiments demonstrate that the proposed algorithm RF-RBM is best suited for cyberattack detection and classification in SCADA systems for smart grids. This is shown by the excellent accuracy, sufficient precision, respectable recall, and a high f1 score demonstrated by the proposed algorithm.

REFERENCES

[1] A. N. Milioudis, G. T. Andreou, and D. P. Labridis, "Enhanced protection scheme for smart grids using power line communications techniques—Part I: Detection of high impedance fault occurrence," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1621–1630, Dec. 2012, doi: 10.1109/TSG.2012.2208987.

[2] C. P. Vineetha and C. A. Babu, "Smart grid challenges, issues and solutions," in *Proc. Int. Conf. Intell. Green Building Smart Grid (IGBSG)*, Apr. 2014, pp. 1–4, doi: 10.1109/IGBSG.2014.6835208.

[3] M. Zhengyou, "Study on the application of advanced power electronics in smart grid," in *Proc. 6th Int. Conf. Future Gener. Commun. Technol. (FGCT)*, Aug. 2017, pp. 1–4, doi: 10.1109/FGCT.2017.8103739.

[4] M. Cao, K. Cao, B. Wu, and M. Tan, "Intelligent condition monitoring and management for power transmission and distribution equipments in Yunnan power grid," in *Proc. Int. Conf. High Voltage Eng. Appl.*, Sep. 2012, pp. 8–11, doi: 10.1109/ICHVE.2012.6357153.

[5] J. Shair, H. Li, J. Hu, and X. Xie, "Power system stability issues, classifications and research prospects in the context of high-penetration of renewables and power electronics," *Renew. Sustain. Energy Rev.*, vol. 145, Jul. 2021, Art. no. 111111.

[6] K. Ullah, A. Basit, Z. Ullah, S. Aslam, and H. Herodotou, "Automatic generation control strategies in conventional and modern power systems: A comprehensive overview," *Energies*, vol. 14, no. 9, p. 2376, Apr. 2021.

[7] Y. Himri, S. M. Muyeen, F. H. Malik, S. Himri, K. A. bin Ahmad, N. K. Merzouk, and M. Merzouk, "A review on applications of the standard series IEC 61850 in smart grid applications," in *Cyberphysical Smart Cities Infrastructures: Optimal Operation and Intelligent Decision Making*. 2022, pp. 197–253.

[8] H. F. Habib, N. Fawzy, and S. Brahma, "Performance testing and assessment of protection scheme using real-time hardware-in-the-loop and IEC 61850 standard," *IEEE Trans. Ind. Appl.*, vol. 57, no. 5, pp. 4569–4578, Sep. 2021.

[9] A. Draz, M. M. Elkholy, and A. A. El-Fergany, "Soft computing methods for attaining the protective device coordination including renewable energies: Review and prospective," *Arch. Comput. Methods Eng.*, vol. 28, no. 7, pp. 4383–4404, Dec. 2021.

[10] Y.-F. Li and C. Jia, "An overview of the reliability metrics for power grids and telecommunication networks," *Frontiers Eng. Manage.*, vol. 8, no. 4, pp. 531–544, Dec. 2021.

[11] Y. Shi, Y. Li, Y. Zhou, R. Xu, D. Feng, Z. Yan, and C. Fang, "Optimal scheduling for power system peak load regulation considering short-time startup and shutdown operations of thermal power unit," *Int. J. Elect. Power Energy Syst.*, vol. 131, Oct. 2021, p. 107012.

[12] A. Oshnoei, M. Kheradmandi, S. M. Muyeen, and N. D. Hatziargyriou, "Disturbance observer and tube-based model predictive controlled electric vehicles for frequency regulation of an isolated power grid," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 4351–4362, Sep. 2021.

[13] D. K. Panda and S. Das, "Smart grid architecture model for control, optimization and data analytics of future power networks with more renewable energy," *J. Cleaner Prod.*, vol. 301, Jun. 2021, Art. no. 126877.

[14] A. Ghasempour, "Internet of Things in smart grid: Architecture, applications, services, key technologies, and challenges," *Inventions*, vol. 4, no. 1, p. 22, Mar. 2019.

[15] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," *Comput. Netw.*, vol. 169, Mar. 2020, Art. no. 107094.

[16] M. Srivastava, "An overview of cyber-security issues in smart grid," in *Computer Networks, Big Data and IoT* (Lecture Notes on Data Engineering and Communications Technologies), vol. 66, A. Pandian, X. Fernando, and S. M. S. Islam, Eds. Singapore: Springer, 2021, pp. 643–650, doi: 10.1007/978-981-16-0965-7_49.

[17] S. Liu, S. You, H. Yin, Z. Lin, Y. Liu, W. Yao, and L. Sundaresh, "Model-free data authentication for cyber security in power systems," *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 4565–4568, Sep. 2020.

[18] P. P. Biswas, H. C. Tan, Q. Zhu, Y. Li, D. Mashima, and B. Chen, "A synthesized dataset for cybersecurity study of IEC 61850 based substation," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2019, pp. 1–7.

[19] C. Mu, T. Ding, M. Qu, Q. Zhou, F. Li, and M. Shahidehpour, "Decentralized optimization operation for the multiple integrated energy systems with energy cascade utilization," *Appl. Energy*, vol. 280, Dec. 2020, Art. no. 115989.

[20] E. Hammad, M. Ezeme, and A. Farraj, "Implementation and development of an offline co-simulation testbed for studies of power systems cyber security and control verification," *Int. J. Electr. Power Energy Syst.*, vol. 104, pp. 817–826, Jan. 2019.

[21] H. Wang, J. Ruan, B. Zhou, C. Li, Q. Wu, M. Q. Raza, and G.-Z. Cao, "Dynamic data injection attack detection of cyber physical power systems with uncertainties," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5505–5518, Oct. 2019.

[22] M. Ghiasi, M. Dehghani, T. Niknam, A. Kavousi-Fard, P. Siano, and H. H. Alhelou, "Cyber-attack detection and cyber-security enhancement in smart DC-microgrid based on blockchain technology and Hilbert Huang transform," *IEEE Access*, vol. 9, pp. 29429–29440, 2021.

[23] Q. Wang, W. Tai, Y. Tang, M. Ni, and S. You, "A two-layer game theoretical attack-defense model for a false data injection attack against power systems," *Int. J. Electr. Power Energy Syst.*, vol. 104, pp. 169–177, Jan. 2019.

[24] A. Khan, M. Hosseinzadehtaher, M. B. Shadmand, D. Saleem, and H. Abu-Rub, "Intrusion detection for cybersecurity of power electronics dominated grids: Inverters PQ set-points manipulation," in *Proc. IEEE CyberPELS (CyberPELS)*, Oct. 2020, pp. 1–8.

[25] B. Hu, C. Zhou, Y.-C. Tian, X. Hu, and X. Junping, "Decentralized consensus decision-making for cybersecurity protection in multimicrogrid systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 4, pp. 2187–2198, Apr. 2021.

[26] H. Kure, S. Islam, and M. Razzaque, "An integrated cyber security risk management approach for a cyber-physical system," *Appl. Sci.*, vol. 8, no. 6, p. 898, May 2018.

[27] R. Lai, X. Qiu, and J. Wu, "Robustness of asymmetric cyber-physical power systems against cyber attacks," *IEEE Access*, vol. 7, pp. 61342–61352, 2019.

[28] T. Bailey, J. Johnson, and D. Levin, "Deep reinforcement learning for online distribution power system cybersecurity protection," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2021, pp. 227–232.

[29] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, "Adversarial attacks on machine learning cybersecurity defences in industrial control systems," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102717.

[30] M. Mohammadpourfard, Y. Weng, M. Pechenizkiy, M. Tajdinian, and B. Mohammadi-Ivatloo, "Ensuring cybersecurity of smart grid against data integrity attacks under concept drift," *Int. J. Electr. Power Energy Syst.*, vol. 119, Jul. 2020, Art. no. 105947.

[31] V. S. B. Kurukuru, M. A. Khan, and S. Sahoo, "Cybersecurity in power electronics using minimal data—A physics-informed spline learning approach," *IEEE Trans. Power Electron.*, vol. 37, no. 11, pp. 12938–12943, Nov. 2022.

[32] Y. Liu, J. Liu, L. Ma, and L. Tian, "Artificial root foraging optimizer algorithm with hybrid strategies," *Saudi J. Biol. Sci.*, vol. 24, no. 2, pp. 268–275, Feb. 2017.

[33] Y. Liu, J. Liu, L. Tian, and L. Ma, "Hybrid artificial root foraging optimizer based multilevel threshold for image segmentation," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–16, 2016.

[34] X. He, H. Chen, B. Niu, and J. Wang, "Root growth optimizer with self-similar propagation," *Math. Problems Eng.*, vol. 2015, pp. 1–12, 2015.

[35] Z. Wang, M. V. Kleunen, H. J. During, and M. J. A. Werger, "Root foraging increases performance of the clonal plant potentilla reptans in heterogeneous nutrient environments," *PLoS ONE*, vol. 8, no. 3, 2013, Art. no. e58602.

[36] L. Ma, K. Hu, Y. Zhu, and H. Chen, "A hybrid artificial bee colony optimizer by combining with life-cycle, Powell's search and crossover," *Appl. Math. Comput.*, vol. 252, pp. 133–154, Feb. 2015.

[37] L. Ma, Y. Zhu, Y. Liu, L. Tian, and H. Chen, "A novel bionic algorithm inspired by plant root foraging behaviors," *Appl. Soft Comput.*, vol. 37, pp. 95–113, Dec. 2015.

[38] M. Kuchhold, M. Simon, and T. Sikora, "Restricted Boltzmann machine image compression," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 243–247, doi: 10.1109/PCS.2018.8456279.

[39] Z. Liu, R. Wang, N. Japkowicz, D. Tang, W. Zhang, and J. Zhao, "Research on unsupervised feature learning for Android malware detection based on restricted Boltzmann machines," *Future Gener. Comput. Syst.*, vol. 120, pp. 91–108, Jul. 2021.

[40] R. W. R. de Souza, D. S. Silva, L. A. Passos, M. Roder, M. C. Santana, P. R. Pinheiro, and V. H. C. de Albuquerque, "Computer-assisted Parkinson's disease diagnosis using fuzzy optimum-path forest and restricted Boltzmann machines," *Comput. Biol. Med.*, vol. 131, Apr. 2021, Art. no. 104260.

[41] L. Xing, K. Demertzis, and J. Yang, "Identifying data streams anomalies by evolving spiking restricted Boltzmann machines," *Neural Comput. Appl.*, vol. 32, pp. 6699–6713, Jun. 2020.

[42] X. Lü, L. Meng, C. Chen, and P. Wang, "Fuzzy removing redundancy restricted Boltzmann machine: Improving learning speed and classification accuracy," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 10, pp. 2495–2509, Oct. 2020.

[43] K. Demertzis, L. Iliadis, E. Pimenidis, and P. Kikiras, "Variational restricted Boltzmann machines to automated anomaly detection," *Neural Comput. Appl.*, vol. 1, pp. 15207–15220, Mar. 2022.

[44] Mississippi State University Critical Infrastructure Protection Center. (Apr. 2014). *Industrial Control System Cyber Attack Data Set*. [Online]. Available: http://www.ece.msstate.edu/wiki/index.php/ICS_Attack_Dataset

[45] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015, doi: 10.1109/TSG.2015.2409775.

[46] S. Pan, T. Morris, and U. Adhikari, "Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 650–662, Jun. 2015, doi: 10.1109/TII.2015.2420951.

[47] R. C. Borges Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2014, pp. 1–8, doi: 10.1109/ISRCS.2014.6900095.

[48] B. Riyaz and S. Ganapathy, "A deep learning approach for effective intrusion detection in wireless networks using CNN," *Soft Comput.*, vol. 24, no. 22, pp. 17265–17278, Nov. 2020.

[49] L. Haghnegahdar and Y. Wang, "A whale optimization algorithm-trained artificial neural network for smart grid cyber intrusion detection," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9427–9441, Jul. 2020.

[50] J. Qian, X. Du, B. Chen, B. Qu, K. Zeng, and J. Liu, "Cyber-physical integrated intrusion detection scheme in SCADA system of process manufacturing industry," *IEEE Access*, vol. 8, pp. 147471–147481, 2020.

[51] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, p. 916, Jun. 2020.

[52] M. Choraś and M. Pawlicki, "Intrusion detection approach based on optimised artificial neural network," *Neurocomputing*, vol. 452, pp. 705–715, Sep. 2021.

[53] G. O. Young, "Synthetic structure of industrial plastics," in *Plastics*, vol. 3, J. Peters, Ed., 2nd ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15–64.

[54] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021, doi: 10.1109/ACCESS.2021.3056407.

[55] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, "Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105082, doi: 10.1016/j.engappai.2022.105082.

[56] X. Li, A. Zheng, X. Zhang, C. Li, and L. Zhang, "Rolling element bearing fault detection using support vector machine with improved ant colony optimization," *Measurement*, vol. 46, no. 8, pp. 2726–2734, Oct. 2013.

[57] O. A. Alimi, K. Ouahada, A. M. Abu-Mahfouz, and S. Rimer, "Power system events classification using genetic algorithm based feature weighting technique for support vector machine," *Heliyon*, vol. 7, no. 1, Jan. 2021, Art. no. e05936, doi: 10.1016/j.heliyon.2021.e05936.

[58] C. L. Huang and J. F. Dun, "A distributed PSO–SVM hybrid system with feature selection and parameter optimization," *Appl. Soft Comput.*, vol. 8, pp. 1381–1391, Sep. 2008.

SAYAWU YAKUBU DIABA (Graduate Student Member, IEEE) was born in Suhum, Ghana. He received the B.Eng. and M.Sc. degrees in telecommunications engineering from the Kwame Nkrumah University of Science and Technology. He is currently pursuing the D.Sc. (Tech.) degree in telecommunication engineering with the University of Vaasa, Finland. He was formerly employed with Electricity Company of Ghana, where he worked as a Power Distribution Specialist for 13 years. His research interests include the use of machine learning in smart grids, developing cyber security algorithms for smart grids SCADA networks, and performance analysis of smart grids. He is also interested in wireless communication and automation.

MIADREZA SHAFIE-KHAH (Senior Member, IEEE) received the first Ph.D. degree in electrical engineering from Tarbiat Modares University, Tehran, Iran, the second Ph.D. degree in electromechanical engineering from the University of Beira Interior (UBI), Covilha, Portugal. He held postdoctoral positions at UBI and the University of Salerno, Salerno, Italy. Currently, he is a Professor (tenure-track) with the University of Vaasa, Vaasa, Finland. He has coauthored more than 500 papers that received more than 12000 citations with an H-index of 62. His research interests include electricity markets, power system optimization, demand response, electric vehicles, price and renewable forecasting, and smart grids. He has won five best paper awards at IEEE conferences. He was considered one of the Outstanding Reviewers of the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, in 2014 and 2017, the IEEE TRANSACTIONS ON POWER SYSTEMS, in 2017 and 2018, and the IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY, in 2020 and 2021; and one of the Best Reviewers of the IEEE TRANSACTIONS ON SMART GRID, in 2016 and 2017. He is a Top Scientist in the Research.com ranking in engineering and technology. He is an Editor of the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY and the IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY; an Associate Editor of the IEEE SYSTEMS JOURNAL, IEEE ACCESS, and *IET-RPG*; the Guest Editor-in-Chief of the IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY; and the Guest Editor of the IEEE TRANSACTIONS ON CLOUD COMPUTING and more than 14 special issues. He is also the Volume Editor of the book titled *Blockchain-Based Smart Grids* (Elsevier, 2020).

MOHAMMED ELMUSRATI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees (Hons.) in electrical and electronic engineering from the University of Benghazi, Libya, in 1991 and 1995, respectively, and the Licentiate of Science degree (Hons.) in technology and the D.Sc. degree in technology, automation and control engineering from Aalto University, Finland, in 2002 and 2004, respectively. He is a Full Professor of communication, automation, and digitalization with the School of Technology and Innovations, University of Vaasa, Finland. He has developed several international programs, such as the Communication and Systems Engineering Program and the Industrial Digitalization Program. Now, he is the Head of the International Program of Sustainable and Autonomous Systems (SAS). He has published about 160 papers, books, and book chapters. His research interests include wireless communications, artificial intelligence, machine learning, biotechnology, data analysis, stochastic systems, and game theory.

Publication IV

Neural Networks 165 (2023) 321–332

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

SCADA securing system using deep learning to prevent cyber infiltration

Sayawu Yakubu Diaba [a,*], Theophilus Anafo [b], Lord Anertei Tetteh [c], Michael Alewo Oyibo [d], Andrew Adewale Alola [e,f], Miadreza Shafie-khah [g], Mohammed Elmusrati [a]

[a] Department of Telecommunication Engineering, School of Technology and Innovations, University of Vaasa, Vaasa, Finland
[b] Department of Electrical/Electronic Engineering, Cape Coast Technical University 5P3+F7H, Cape Coast, Ghana
[c] Department of Electrical and Electronic Engineering, Koforidua Technical University, 3P8P+F5F, Koforidua, Ghana
[d] National Bureau of Statistics, Abuja, Nigeria
[e] CREDS-Centre for Research on Digitalization and Sustainability, Inland Norway University of Applied Sciences, Norway
[f] Faculty of Economics, Administrative and Social Sciences, Nisantasi University, Istanbul, Turkey
[g] Department of Electrical Engineering, School of Technology and Innovations, University of Vaasa, Vaasa, Finland

ARTICLE INFO

Article history:
Received 15 February 2023
Received in revised form 14 May 2023
Accepted 23 May 2023
Available online 2 June 2023

Keywords:
Genetically seeded flora
Intrusion detection systems
Long short-term memory
Recurrent neural network
Residual neural network
And transformer neural network

ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems are computer-based control architectures specifically engineered for the operation of industrial machinery via hardware and software models. These systems are used to project, monitor, and automate the state of the operational network through the utilization of ethernet links, which enable two-way communications. However, as a result of their constant connectivity to the internet and the lack of security frameworks within their internal architecture, they are susceptible to cyber-attacks. In light of this, we have proposed an intrusion detection algorithm, intending to alleviate this security bottleneck. The proposed algorithm, the Genetically Seeded Flora (GSF) feature optimization algorithm, is integrated with Transformer Neural Network (TNN) and functions by detecting changes in operational patterns that may be indicative of an intruder's involvement. The proposed Genetically Seeded Flora Transformer Neural Network (GSFTNN) algorithm stands in stark contrast to the signature-based method employed by traditional intrusion detection systems. To evaluate the performance of the proposed algorithm, extensive experiments are conducted using the WUSTL-IIOT-2018 ICS SCADA cyber security dataset. The results of these experiments indicate that the proposed algorithm outperforms traditional algorithms such as Residual Neural Networks (ResNet), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) in terms of accuracy and efficiency.

1. Introduction

Industry 4.0, also known as the Fourth Industrial Revolution (Smith & Fressoli, 2021), began in the early 2000s (Montalban, Iradier, & Member, 2020) as a result of advancements in internet communication and the development of automated software and frameworks (Hoffmann Souza, da Costa, de Oliveira Ramos, & da Rosa Righi, 2021). This has made it possible to control the manufacturing process using simple computer programs and microcontrollers, leading to increased customization of products (Chen & Chang, 2020; Rousopoulou et al., 2022). Research is ongoing to develop self-decision-making control systems for manufacturing processes and to enable remote monitoring and control of these processes (Hassan Malik, Alam, Kusik, & Moullec, 2020; Jasperneite, Sauter, & Wollschlaeger, 2020; Sarker, 2022). SCADA systems are mainly used to control and monitor (Kumar & S, 2020), components of vital infrastructures (Kirubakaran, 2020), such as smart grids, pipelines, transportation, telecommunication, and manufacturing plants (Lee & Hong, 2020). The SCADA systems can also act as a status projector for monitoring the operation of the Industrial Control Systems (ICS). It can be integrated with a Programmable Logic Controller (PLC) and other control technologies like Proportional Integral Derivative (PID) controllers (V, 2020). SCADA devices have a high operational speed, enabling real-time data analysis.

The high integration of communication infrastructure in the smart grid and the connections to the internet (Cherifi & Hamami, 2018; Yang, McLaughlin, Sezer, Yuan, & Huang, 2014) in the

* Corresponding author.
E-mail address: sdiaba@uwasa.fi (S.Y. Diaba).
URL: http://dx.doi.org/10.1016/j.cviu.2017.00.000 (S.Y. Diaba).

https://doi.org/10.1016/j.neunet.2023.05.047

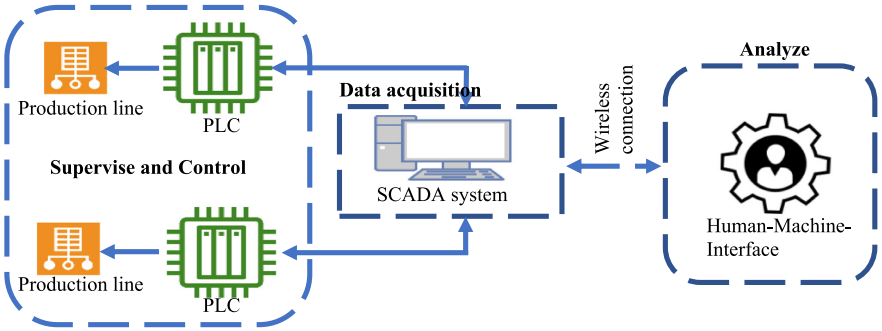


Fig. 1. The architectural overview of a SCADA system.

Table 1
Comparison between SCADA and IoT.

Parameters	SCADA	IoT
Communication medium	Semi-wireless	Wireless
Storage	Local	Cloud
System integration	Limited integration to the peripheral	Easily integrate with the peripheral
Operational reliability	High	Low
Suitability	Suitable for a big production line	Suitable for minor applications
Threat possibility	Medium to high	High

SCADA system have created a lacuna for cyber-attacks (Altunay, Albayrak, Ozalp, & Cakmak, 2021; Singh, Ebrahem, & Govindarasu, 2019). A technology designed with the sole purpose of granting cyber protection for computers, computer networks, data transmissions, and legitimate access is termed *Cybersecurity*. The cyber-security systems are primarily a setup of computer (host) security systems and network security systems. Each of these has, at minimum, antivirus software, a firewall, and Intrusion Detection System (IDS) (Singh, Garg, Kumar, & Saquib, 2015).

SCADA system security concerns are receiving more attention as the frequency of security incidents against these crucial infrastructures is rising (Samdarshi, Sinha, & Tripathi, 2016). Though, the presence of cyber threats in SCADA systems are comparatively less compared to the Internet of Things (IoT) systems because the SCADA networks are not connected to an open internet as the IoT systems (see Fig. 1).

Yet, the SCADA systems are in the third top position in terms of attaining frequent threat disturbances amongst the other applications. The cyber-attacks are targeted at the SCADA systems for manipulating the operational control of the network. The hackers may cause damage to the power system when they have control over the switches, isolators, and relays under the command of the SCADA systems (see Table 1).

These attacks' repercussions may jeopardize the safety, availability, reputation, profitability, and reliability of the targeted organizations (Liu & Wang, 2022). Traditional SCADA systems were not created with a cyber-securing protocol, albeit at the moment some models do have certain firewall security measures activated. The SCADA systems are still a target for hackers (Altaha, Lee, Aslam, & Hong, 2020) who exploit the firewall's weaknesses. To protect the communication infrastructure of the smart grid, it is essential to develop a SCADA network intrusion detection solution that considers both operational requirements and particular traffic characteristics of SCADA systems (see Table 2).

Motivated by the above facts, the following are regarded as the encapsulation of the primary contributions of this paper:

- We investigate the application of deep learning techniques in detecting cyber threats in both industrial and general settings with a specific focus on SCADA systems used in the smart grid. We explore the potential of these techniques in identifying and mitigating cyber-attacks in various environments, highlighting their effectiveness and limitations. We also investigate the possible integration of deep learning techniques with existing security systems to enhance their performance and overall security.
- We propose an algorithm for detecting cyber intrusions by analyzing changes in operational patterns that are related to intrusion activity. To achieve this, we utilize a GSFTNN algorithm that is custom-made for the specific task of intrusion detection. The algorithm is designed to identify anomalies in the operational patterns and flag them as potential intrusions.
- We perform extensive simulations using the WUSTL-IIOT-2018 ICS SCADA cyber security dataset to evaluate the effectiveness of deep learning techniques in detecting cyber-attacks in SCADA systems. The simulation process consists of two stages: binary classification and multiclass classification. In the binary classification stage, the data is classified as normal and attacks. In the multiclass detection stage, the data is further classified into three categories: exploiting attacks, aggressive attacks, and normal traffic. The results of the simulation are used to analyze the effectiveness of the deep learning techniques and to pinpoint possible areas for development.

The remainder of the paper is structured as follows; related studies are presented in Section 2. The methodology is in Section 3 where we give the data description as well as the types of attacks and the summary of the attacks therein. In Section 4, experimental analysis is presented and finally, the paper's conclusion is presented in Section 5.

2. Related studies

In the area of SCADA security, older papers have investigated the use of machine learning techniques to enhance security. For example, the authors of Maglaras and Jiang (2014) proposed a machine learning-based approach for detecting anomalies in SCADA systems and compares the performance of several different algorithms, including neural networks, support vector machines, and decision trees. The authors proposed a novel method for detecting intrusions in the SCADA system, which can identify abnormal activity even if an attacker attempts to conceal it in the control layer of the system. To assess the effectiveness of the algorithms, supervised machine learning models were examined to categorize normal and abnormal behaviors in an ICS. The authors

Table 2
Some types of attacks involved in the SCADA system.

Attack types	Reflection	Initiation
Denial of Service	Enforcing maximum traffic to the network to block the actual communication	Poor authentication platform
Ransomware attacks	Malfunction and operational block of PLCs	Vulnerable hardware
Malicious node attacks	Execution of unauthorized operation	Web interface with an outdated operating system
Phishing attacks	Control over the SCADA system	Absence of network isolation and weak authentication
Worm attacks	Blocks access/operation	No network isolation
Honeypot attacks	Reframe the device function	Weak servers and vulnerable policies on security

used several machine learning models in examining the models, and they performed well at spotting abnormalities, particularly stealthy attacks. According to the findings, random forest outperforms other classifier algorithms (Mokhtari, Abbaspour, Yen, & Sargolzaei, 2021).

In Lopez Perez, Adamsky, Soua, and Engel (2018) a machine learning approach for intrusion detection in SCADA systems was accessed on a real-world dataset. The authors find that the random forest detects intrusion effectively. Older papers demonstrate that machine learning has been a topic of interest in the area of SCADA security for at least a decade and that the use of machine learning for enhancing SCADA security is not a new idea. The authors of Teixeira et al. (2018) looked at cyber-attacks that use AI-based techniques and found some mitigation techniques that can be used to stop such attacks. Also, they examined current trends in AI-based cyber-attacks and were able to identify the methodologies and strategies currently used in executing AI-based cyber-attacks as well as what future scenarios will likely be conceivable to control such attacks.

Several studies have investigated the use of artificial neural networks (ANN), convolutional neural networks (CNN), and RNN to detect and prevent cyber-attacks in SCADA systems. These methods have been demonstrated to be successful in detecting and preventing a wide range of cyber-attacks, including malware, phishing, and distributed denial-of-service attacks (Al Husaini, Habaebi, Hameed, Islam, & Gunawan, 2020; Balla, Habaebi, Islam, & Mubarak, 2022; Khan, Zhang, Alazab, & Kumar, 2019). A CNN (P, Hong, Gao, Yao, & Zhang, 2020; Wu, Hong, & Chanussot, 2022) defining the significant temporal patterns of SCADA communication and pinpointing time windows that are vulnerable to network attacks rather than hand-crafted characteristics for specific network packets or flows was proposed. The authors provided a re-training method to manage instances of network attacks that have never been detected before. The study utilized actual SCADA traffic datasets and the results demonstrate that the deep-learning-based technique that has been proposed is suitable for SCADA systems' network intrusion detection, attaining high detection accuracy and offering the capacity to address newly emerging threats (Altaha et al., 2020; Yang, Cheng, & Chuah, 2019).

The use of deep learning for enhancing the security of SCADA systems has been a growing area of research in recent years. Studies that focus on deep learning, suggest that this area of research has advanced significantly (Gao et al., 2023; Wu, Hong, & Chanussot, 2023), and that deep learning is a promising direction (Yang & Chen, 2019) for enhancing the security of SCADA systems. With the increasing use of technology in critical infrastructures, such as medical devices, power plants, and water treatment facilities (Lee & Hong, 2020; Pliatsios, Sarigiannidis, Lagkas, & Sarigiannidis, 2020), the need for robust and secure SCADA systems is more pressing than ever. Cyber-attacks on SCADA systems can result in significant harm, including disruption of essential services, loss of sensitive information, and physical damage to equipment. To address these concerns, many researchers have turned to deep learning as a promising solution for enhancing the security of SCADA systems (Avola, Cinque, Fagioli, & Foresti, 2022).

The research in Wang, Harrou, Bouyeddou, Senouci, and Sun (2022) presented a stacked deep learning-driven method for detecting cyber-attacks. The relevant aspects of the suspicious behaviors were thoroughly learned by the proposed stacked deep learning model, which then distinguishes them from normal actions. As a result, the stacked deep learning-based intrusion detection approach performs better than some cutting-edge shallow methods, such as the standalone deep learning models, naive Bayes, random forests, nearest neighbor, oneR, AdaBoost, and support vector machine. The research in Jmila and Houda (2022) focuses more on shallow classifiers, which are still often employed in machine learning-based IDS because of their maturity and ease of usage. The authors tested the resistance to various adversarial approaches often utilized in the state-of-the-art of AdaBoost, bagging, decision tree, gradient boosting, logistic regression, random forest, support vector classifier, and even a deep learning network. A Gaussian data augmentation defensive method was implemented and its impact on increasing classifier robustness was assessed. The findings demonstrate that not all classifiers are affected equally by attacks, that a classifier's robustness relies on the attack, and that depending on the network intrusion detection scenario, a trade-off between performance and robustness must be considered.

It is worth noting that while deep learning has shown great promise in enhancing SCADA security, there are still many challenges to overcome. For example, deep learning models can be vulnerable to adversarial attacks (Jmila & Houda, 2022; Ozdag, 2018) and the quality of training data can significantly impact the performance of these models. Nevertheless, the research in this area suggests that deep learning is a promising direction for enhancing the security of SCADA systems. It has the potential to enhance the security of SCADA systems in a variety of ways, including detecting and preventing cyber-attacks, mitigating system failures, protecting sensitive information, and enhancing the security of communication networks. However, as with any new technology, there are still many challenges to overcome, such as improving the robustness of deep learning models, addressing the issue of data scarcity, and developing secure deep learning systems that are resistant to adversarial attacks.

3. Methodology

Addressing cyber intrusion in SCADA is the main motive of the proposed algorithm and it is implemented in this paper with a deep learning-based approach. The algorithm is a hybrid of GSF and TNN and it is compared to ResNet, RNN, and LSTM models for identifying the best-performing algorithm in detecting intrusions in SCADA systems. Washington University St. Louis-Industrial IoT-2018 (WUSTL-IIOT-2018) dataset for ICS SCADA cybersecurity used in Ahakonye, Nwakanma, Lee, and Kim (2023) is the dataset used in this study to examine the efficiency and accuracy of the above-mentioned algorithms.

3.1. Data

The WUSTL-IIOT-2018 ICS SCADA is a collection of network traffic data captured from a real-world ICS that was intentionally

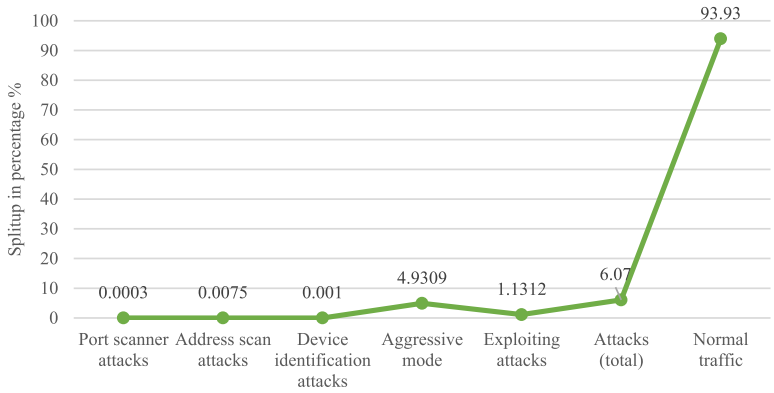


Fig. 2. The summary of attacks available in the dataset.

subjected to cyber-attacks. The dataset was created to evaluate intrusion IDS in cyber–physical systems. It was collected from a water treatment testbed in the United States, a representative example of a real-world industrial control system. To generate the intrusion data, the testbed is connected to a network monitoring system accompanied by a scan tool, and its features are summarized to form a dataset. The following are the attacks available in the dataset along with their data generation procedure.

3.1.1. Port scanner attack

The port scanner attacks are included in the SCADA system for observing its active ports in the operation and control process. To generate such attacks, certain targeted nodes are generated at different frequencies of time using the Nmap tool. At the same time, the Transmission Control Protocol (TCP) connection is also partially disabled for allowing the attack to generate the data attributes in the testbed.

3.1.2. Address scanner attack

The address scan attacks are generated to observe the Modbus server address. This allows the attacker to reach the connected hardware devices of the SCADA system for implementing the malfunctioning algorithm. In general, the SCADA systems are engaged with only one Modbus address and it gives an open path to the intruder for generating different types of attacks when it is tracked.

3.1.3. Device identification attack

The attackers are creating the device identification attack to find out the specification and model numbers of the connected devices to the SCADA network. It can be produced by tracking the Modbus slave identification of the targeted SCADA system. Thus, the vulnerable hardware devices must be verified regularly in the SCADA systems. In some cases, device identification attacks are avoided by having an additional authentication process.

3.1.4. Aggressive model device attack

In the aggressive mode of device identification attack, the information of all the slave buses is collected along with the Modbus slave identification. These kinds of attacks are employed in the SCADA network to freeze all the connected hardware without sending any malicious nodes. In real-time applications, each connected hardware is implemented with a separate authentication process. This improves the complexity of the security algorithm and restricts the success rate of aggressive mode attacks.

Table 3
Sample of the employed dataset.

Sport	Tpkt	Tbyte	Spkt	Dpkt	Sbyte	Tgt
143	2	180	2	0	180	0
68	2	684	2	0	684	0
0	1	60	1	0	60	0
61 845	20	127	10	10	644	0
61 846	20	127	10	10	644	0
44 287	6	372	4	2	248	1
48 456	20	128	12	8	776	1
48 458	20	139	12	8	782	1
44 460	20	128	12	8	776	1
61 850	12	780	6	6	396	0
61 849	12	780	6	6	396	0
61 848	18	1152	10	8	644	0

3.1.5. Exploit attack

In exploit attacks, information about the operational state of PLC coils is obtained to understand how the SCADA system is currently functioning. This allows attackers to replicate the manufacturing process and produce identical products. A system for generating and using inspection records was used to observe the movements of normal and malicious nodes in a testbed model during the dataset generation process. The testbed model was made to run continuously for 25 h to monitor the changes in the network. Fig. 2 presents the summary of attacks available in the dataset utilized.

Table 3 includes several features that describe various aspects of network communications. One of these features is the source port (sport), which represents the number of unique source ports. Another feature is the total packets (TotPkt), which represent the total number of packets involved in the communications. Additionally, the total bytes (TotBytes) feature indicates the total number of bytes transferred. Two other features included in Table 3 are the source packets (SrcPkts) and destination packets (DstPkts). These features represent the number of packets transmitted from the source and the number of packets received at the destination, respectively. Finally, the Source bytes (SrcBytes) feature indicates the number of bytes transferred from the source to the destination during communications. Together, these features provide a comprehensive picture of the different aspects of network communications that can be used to analyze and understand network traffic patterns (see Fig. 3).

3.2. Preprocessing

The total number of traffic data available in the dataset is 7 037 983 counts. In that 427 206 instances are attack-oriented

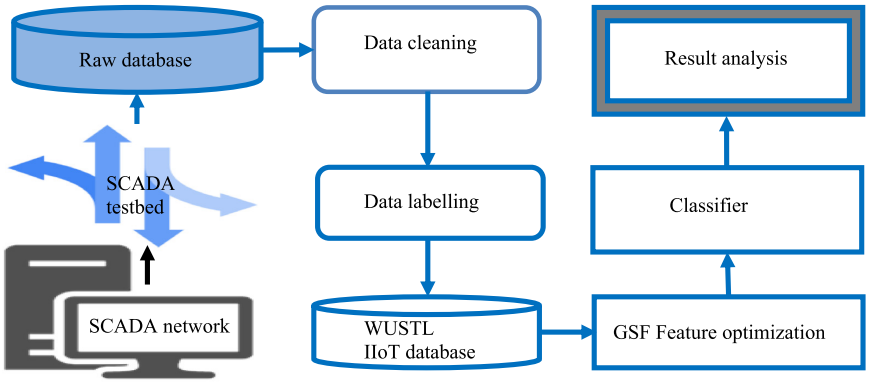


Fig. 3. The design display of the proposed model.

Table 4
Summary of the dataset.

Data type	Timetable
Number of variables	6
Number of rows	7 037 983
Number of variables with missing	0
Number of variables with duplicate	6
Timestamp is regularly spaced	True
Timestamp has missing	False
Timestamp has duplicates	False
Timestamp sorted	True

and the remaining 6 610 777 belong to the normal traffic category. In the proposed algorithm, a dataset split ratio of 70:30 is employed after cross-validation of five folds is implemented in the phase 1 simulations, the proposed algorithm is utilized to identify the normal and attack traffics, and in phase 2, the proposed algorithm was experimented to categorize the traffics as normal traffic, exploiting attacks and aggressive mode attack.

The workflow followed in the proposed algorithm does not have any pre-processing step as the data available in the dataset are already pre-processed using the data cleaning and labeling process. However, the raw data values collected from the network monitoring tools may have some missing and error data due to sudden fluctuations in its operation. Therefore, a data cleaning (data cleaner app in MATLAB 2022b) process was used to manipulate the missing values in the dataset. The data cleaner app in MATLAB 2022b is an interactive tool for locating messed-up column-oriented data, cleaning numerous variables of data at once, and improving the cleaning process. A total of 7 037 983 samples and 6 variables dataset was loaded into the data cleaner app. We set the data cleaner app to use only standard indicators to detect missing values such as not a number (NaN), not a time (NaT), and cell of character vectors. The remove missing method is used to remove the data rows with missing entries. The outliers are another type of error usually present in the data with an unusual entry. To handle it, we used the fill outlier cleaning method with linear interpolation as the filling method. The method of detection is the moving mean and the threshold was set at 3. Tables 4 and 5 explore the data summary after processing.

3.2.1. Feature optimization

The feature optimization process is employed in this work to handle the abnormalities in the available dataset. In some cases, the feature attributes may remain almost the same on different attack data. Hence it worsens the misclassification and reduces the precision level of the classifier system. A GSF feature optimization algorithm is utilized in the proposed algorithm to avoid such limitations. GSF is an upgraded version of an artificial flora optimization technique that selects the connecting point relevancy based on the seed-growing property of the respective points. The GSF model is equipped with a genetic algorithm for estimating the best seed-growing points. The genetic algorithm estimates the location by analyzing the propagation distance among the points along with the plant weights. The propagation distance d_y of the seeds is predicted using the equation written in (Cheng, Wu, & Wang, 2018; Selvarajan, Shaik, Ameerjohn, & Kannan, 2020).

$$d_y = d_{y1}\,(\psi \times j_1) + d_{y2}\,(\psi \times j_2) \tag{1}$$

where j_1 and j_2 stand for searching coefficients. The uniform random numbers between 0 and 1 are generated by *rand* and denoted by ψ. The grandparent's propagation distance and the parent's propagation distance are denoted by d_{y1} and d_{y2} respectively. The two main steps of the flora optimization technique are the spreading and selection behavior. Thus, the position of the plant is determined using the matrix $P_{i,y}$, the dimension is denoted by i and y represents the total number of plants in the flora. The equation for the spreading process can be written as

$$P_{i,y} = \psi \times d\,(2 - d) \tag{2}$$

where d is for the maximum limit area. Since the weight value may be determined by the standard deviation of the propagation distance between the parent plant and offspring plant when updating the plant position, we can express the equation as

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}\left(P_{i,y} - P'_{i,y}\right)^2}{N}} \tag{3}$$

The present position of the offspring plant is estimated as

$$P'_{i,y^*j} = D_{i,y^*j} + P_{i,y} \tag{4}$$

where, the position of the original plant is denoted by $P_{i,y}$, j is the maximum number of seeds that a single plant can produce, D_{i,y^*j} is for the random estimation of Gaussian distribution value with zero mean and j variance. The final best estimation of the offspring plant is estimated by the probability of survival given as

$$P = \left|\sqrt{\frac{F\left(P'_{i,y^*j}\right)}{F_{\max}}}\right| * Q_x^{y^*j-1} \tag{5}$$

Table 5
Data summary after preprocessing.

	Missing count	Minimum	Maximum	Mean	Median	Mode	Standard deviation
Time	0	00:00:00	69:46:38	34:53:19	53.09	00:00:00	20 000
Sport	0	0	42 238	0.00003589	35 458	354	0.00621
TotPkt	0	1	96 057	9.4518	2	2	66 196
TotBytes	0	60	98 180 744	0.0003317	124	124	4.156e−5
SrcPkts	0	1	96 057	8.0302	2	2	
DstPkts	0	0	4881	1.6534	0	0	35.7538
SrcBytes	0	60	6 942 046	915.1435	124	124	4.81e−4

The selective probability is represented by Q_x and the value of Q_x must fall within the range of 0 and 1. According to the authors of Cheng et al. (2018), having a higher value of Q_x is desirable for problems that can easily get into a local optimal solution. P'_{i,y^*j} represents the fitness of the jth solution. F_{max} represents the flora's maximum fitness. The best characteristics among the overall attributes are chosen from this phase and sent to the classification step. In doing so, the classifier automates the rule-generation process to predict the class label. This improves classification accuracy.

3.3. Classifier

Classifiers are like algorithmic filters used to segregate the given data samples into their respective classes based on the instructions learned from their training samples. The learning process of a classifier model can be refined by implementing a customized preprocessing or feature selection model. The neural network algorithm assigns the testing sample into a particular class based on their similarity score calculated from the comparison. The proposed algorithm explores the following classifiers on the SCADA cyber-attack dataset to find the most suitable model for the real-time application. Classifiers are used to classify data samples into different classes based on the instructions learned from the training samples. Many different types of classifiers can be used for various applications. Some common types include:

Decision Trees: These classifiers use a tree-like structure to make decisions. Each internal node represents a feature of the input data, and each leaf node represents a class label. The algorithm starts at the root node and follows the branches based on the feature values of the input data until a leaf node is reached, which determines the class label of the input.

Naive Bayes: This is a probabilistic classifier that makes class predictions based on the probability of each class given the input features. The "naïve" part of the name refers to the assumption that the features are independent of each other, which is not always true in real-world data.

Neural Networks: These classifiers use a network of artificial neurons to make class predictions. The neurons are organized into layers, with the input layer receiving the input features, one or more hidden layers processing the information, and the output layer producing the class predictions. The network learns to make accurate predictions by adjusting the weights of the connections between the neurons.

Random Forest: This ensemble technique combines different decision trees to make class predictions. Each tree is trained using a different random subset of the input data, and the consensus of all the trees is used to get the final prediction.

Support Vector Machine: This is a type of linear classifier that finds the best boundary (or hyperplane) to separate the input data into different classes. The algorithm is based on finding the line that maximizes the margin, which is the distance between the boundary and the closest points from each class.

In the case of SCADA cyber-attack dataset, it is crucial to find the most suitable model that can quickly and accurately identify cyber-attacks in real-time. The proposed algorithm may compare the performance of these different classifiers on the dataset and select the one that achieves the highest accuracy or lowest false-positive rate.

3.4. Transformer neural network

The TNN algorithm was proposed in the year 2017 to overcome the limitation of computational complexity in many neural network algorithms. It is achieved by utilizing the Graphic Processing Unit (GPU) sources effectively by processing the input data simultaneously. Therefore, the time required for the training process is also limited in the TNN. TNNs are structured with a multi-headed attention layer for learning the input data that allows processing the data in a parallel process. However, in the traditional RNN, the values are considered in sequential order. The TNN is very efficient in natural language processing problems and data mining problems, the architecture is shown in Fig. 4. The encoder and decoder are the two major blocks involved in the TNN architecture and it has a positional encoding block right in front of the encoder block. The role of the positional embedding block is to determine the value of the inputs denoted by (x) at different attributes' places.

In the proposed algorithm the features are counted from '$F_1 \ldots F_5$'. The value of x at F_1may not have the same weight at F_3. The values on certain features may remain the same even if the output classes are different. The positional encoding model addresses such issues and assigns the weights of x into a unique parameter while storing it in the neurons of the TNN. The encoder consists of multi-head attention and a feed-forward block, where the multi-head attention (Selvarajan et al., 2020) has a pair of sub-layers. The input parameters are learned by the multi-head attention block in terms of queries, keys, and range format. The collected parameters are operated with a learnable linear transformation for n times. A constant value is applied in this block as a tuning parameter for operating it to the product of the query with all keys. The output values from the blocks are observed from a SoftMax function from the value of its corresponding weight. The attention outputs are linearly gathered to form a final output where a normalization step is added. Hence the residual connection of the input data is estimated.

$$Q_{ini} = xW^{Q}_{inp} \tag{6}$$

$$K_{inp} = xW^{K}_{inp} \tag{7}$$

$$V_{inp} = xW^{V}_{inp} \tag{8}$$

where, x represents the input, and W represents the customized constant value of the input. The query, keys, and value parameters of the input are represented by Q, K, and V respectively.

$$head_{inp} = \alpha\left(Q_{inp} * K_{inp} * V_{inp}\right) \tag{9}$$

$$\alpha\left(Q_{inp} * K_{inp} * V_{inp}\right) = softmax\left(\frac{Q_{inp} * K^{T}_{inp}}{\sqrt{d}}\right) V_{inp} \tag{10}$$

Algorithm of the GSF feature model:

input = data features (F_1 to F_5) extracted from the testbed
output = selected index of the attributes, F_{sel}
flora seed initialization, F_{si} = rand [size (F_1 to F_5)]
seed-in location of the plants, $P_{i,j} = F_{si}$ (d) (2-d)
propagation distance is taken from d_y
X, Y are representing the feature sizes
Initialize j = 1
  for i = 1 to X*Y
    offspring plants are created by $P_{i,y*j}$
    calculate fitness function
probability estimation on the best plant is observed from 'P'
    when P < rand, then
      F_{sel} (i) = P
reproduction and mutation
j = j + 1
    else
      $P_{i,j} = P_{i,j-1}$
    end if population replacement
end 'i' loop

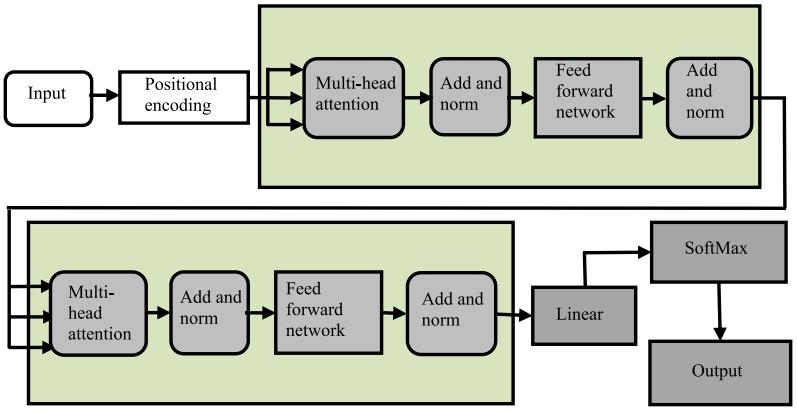


Fig. 4. The architecture of the TNN.

$$norm = x + head_{inp} \tag{11}$$

$$head^2_{inp} = norm\ \left(x + head_{inp}\right) \tag{12}$$

The output attributes obtained from the multi-head attention blocks are moved to a feed-forward network (FFN) for observing an improved output. The normalization process is also included in the output of FFNs and the value of FFN is analyzed as follows

$$FFN = \gamma\ (\delta)\, W_1 + b_1 \tag{13}$$

$$x_{out} = norm(head^2_{inp} + FFN) \tag{14}$$

where γ represents the rectified linear product and δ is for the gated recurrent block.

3.5. Recurrent neural network

The RNN models were developed to regularize the data movement inside the neural network. In the traditional neural network models, the input parameters were allowed to move from one neuron to another neuron without considering anything. As a result, some neurons are unaware of the status of other attributes taken from the input. The RNN regularizes this by making all the attributes follow a sequence movement inside the neural networks. The involvement of the hidden layer makes the neurons store the hidden information regarding the previous attributes, so a small amount of data storage is allocated to each neuron. In some cases, the RNN models are implemented with more than one hidden layer block. There the weight and bias of each hidden will get change from each other to store the different feature information from the given input. Hence the layers included between the input and output layers are independent and do not consider the formation of other hidden layers. The independence of hidden layer weights and biases is making the RNN more complex than their previous models. In some applications, the weights and biases are regularized with the same value, improving computational efficiency. The current state of the neurons is analyzed by

$$Cur_s = f\ (Cur_{s-1}, Inp_s) \tag{15}$$

where Cur_s represents the present state and Inp_s denotes the input state. The activation function of the current state is applied as Eq. (15) and the output is predicted by Eq. (16).

$$Cur_s = \tanh\left(W_{rec} Cur_{s-1} + W_{inp} Inp_s\right) \tag{16}$$

$$Out_s = \tanh\left(W_{out} Cur_s\right) \tag{17}$$

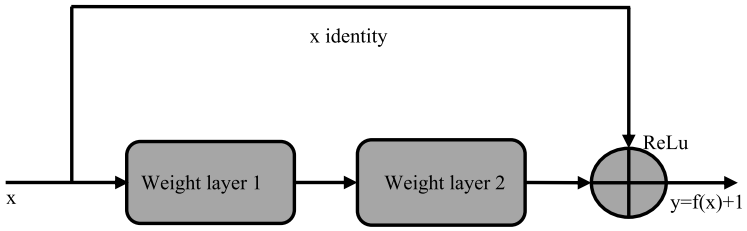


Fig. 5. The architecture of a residual learning.

3.6. Long short-term memory

The LSTM network (Karim, Fazle, Majumdar, & Darabi, 2019) was developed to address the vanishing and exploding gradients problems found in RNN. The LSTM networks were trained to erase the irrelevant information stored in the neuron from the given input. It is achieved by implementing the network with a customized activation function called gates. The internal cell state of the neurons is having the useful information extracted from the training data which is required for the upcoming operation. The LSTM network reads the state of the input gate, modulation gate, forget gate, and output gate by calculating it with element-wise multiple vectors of the given input. Then the neurons will erase the information of the forget gate and combine the information of the input and modulation gate to form output as

$$Cur_s = \left(G_{inp} * G_{inp}^{mod}\right) + \left(G_{for} * Cur_{s-1}\right) \quad (18)$$

where G_{inp} takes gate input, G_{inp}^{mod} stands for gate-modulated input and it is G_{for} for forget gate

3.7. Residual network

ResNet models were proposed to observe more complex features from the given input attributes with a greater number of hidden layers. Each layer in the ResNet is allowed to take some specific feature from the given input. The main idea behind ResNet is to allow the network to learn residual functions, rather than learning the full mapping from input to output. In a residual network, each layer has a shortcut connection that bypasses one or more layers and directly connects the input of the current layer to the output. This allows the network to learn the residual, or the difference between the input and the output of the layer, which is easier to optimize than the full mapping. The residual functions are then added to the output of the corresponding layer, allowing the network to effectively learn the full mapping. ResNet has shown impressive results on a wide range of computer vision tasks, including image classification, object detection, and semantic segmentation. Its architecture has inspired many subsequent neural network models and has become a benchmark for deep learning research (He, Zhang, Ren, & Sun, 2016a, 2016b). However, in some cases, the ResNet was giving poor accuracy by having some unwanted features in its operations. It is addressed by the recent year ResNets by adding dropout and regularization blocks. The architectural view of the ResNet is shown in Fig. 5 with its operational outcome.

4. Experimental analysis and discussion

The experimental work is performed in two phases, binary classification, and multiclass classification. In binary classification, the given information is segregated as normal and attacks. In multiclass detection, the data are classified as exploiting attacks, aggressive attacks, and normal traffic. The performances of the

Table 6
The hyperparameter setting.

Parameter	Total
Validation scheme	Cross-validation
Cross-validation folds	5 folds
Epoch	1000
Activation functions	Softmax, ReLu
Maximum number of split	100
Split criterion	Gini's diversity index

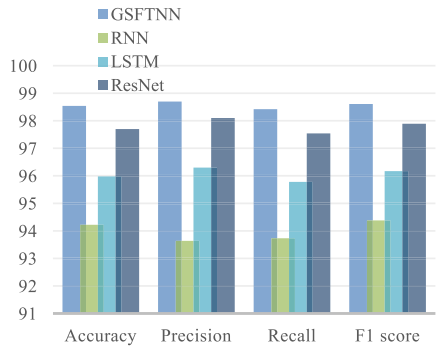


Fig. 6. Phase 1 performance analysis on the verified algorithms.

GSFTNN, RNN, LSTM, and ResNet models are verified with their accuracy, precision, recall, detection rate, and f1 score in both phases (see Tables 6 and 7).

Tables 8 and 9 are representing the performances of the verified algorithms in phase 1 and phase 2 respectively.

The results presented in Fig. 6 demonstrate that the proposed GSFTNN model achieves a high accuracy of 98.54%, indicating its ability to correctly classify a significant majority of the cases. In comparison, the RNN model achieves an accuracy of 94.22%, while the LSTM and ResNet models exhibit accuracy rates of 95.98% and 97.7%, respectively. The proposed GSFTNN model also shows an average precision of 98.7% across all normal and attack categories, outperforming the RNN (93.64%), LSTM (96.3%), and ResNet (98.1%) models. The recall values were 98.42% (Proposed GSFTNN), 93.73% (RNN), 95.78% (LSTM), and 97.54% (ResNet). The F1 score provides a well-rounded evaluation of system performance. In the case of the proposed GSFTNN model, the F1 score is reported as 98.61%, with the RNN, LSTM, and ResNet achieving scores of 94.38%, 96.17%, and 97.89%, respectively.

The result depicted in Fig. 7 indicates the proposed GSFTNN model attained an accuracy of 99.12%, outperforming the comparative models, which attained accuracies of 96.4% (RNN), 97.25% (LSTM), and 98.1% (ResNet), respectively. Additionally, the proposed model attained an average precision of 99.26%, while the precision values for RNN, LSTM, and ResNet were 96.82%, 97.57%, and 98.33%, respectively. The recall values were 98.85% (Proposed

Table 7
Data split up into phases.

Phase 1 – (2 Class)			Phase 2 – (3 Class)		
Classes	Training	Testing	Classes	Training	Testing
Normal traffic	3 966 466	2 644 311	Normal traffic	3 966 466	2 644 311
Attacks	256 324	170 882	Exploiting attacks	47 769	31 845
			Aggressive mode attacks	208 222	138 814

Table 8
Performance of the verified algorithms on the phase 1 dataset.

Algorithms	Accuracy	Precision	Recall	F1 score
GSFTNN	98.54	98.7	98.42	98.61
RNN	94.22	93.64	93.73	94.38
LSTM	95.98	96.3	95.78	96.17
ResNet	97.7	98.1	97.54	97.89

Table 9
Performance of the verified algorithms on the phase 2 dataset.

Algorithms	Accuracy	Precision	Recall	F1 score
GSFTNN	99.12	99.26	98.85	99.2
RNN	96.4	96.82	93.73	96.5
LSTM	97.25	97.57	96.97	97.41
ResNet	98.1	98.33	97.74	98.26

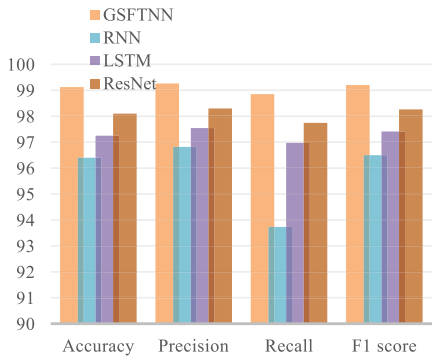


Fig. 7. Phase 2 performance analysis on the verified algorithms.

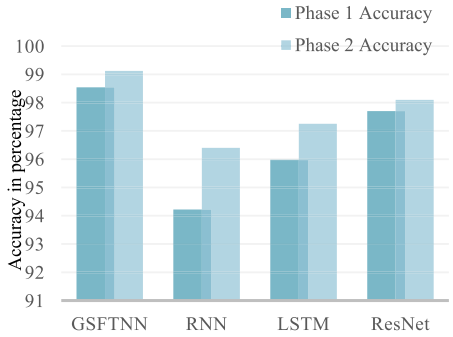


Fig. 8. Comparative analysis of accuracies at both phases.

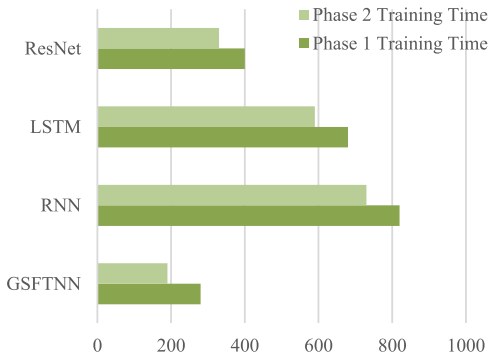


Fig. 9. Comparative analysis of training time with both phases.

GSFTNN), 93.73% (RNN), 96.97% (LSTM), and 97.74% (ResNet). Finally, the GSFTNN model achieved an F1 score of 99.2%, while the RNN, LSTM, and ResNet models scored 96.5%, 97.41%, and 98.26%, respectively.

Fig. 8 indicates the accuracy comparison between the phase 1 and the phase 2 analyses. The phase 2 accuracies are comparatively high in all the algorithms because the attack classes in phase 2 contain only 2 attacks but in phase 1 the data attack model count is 5. The two major attack classes are considered in phase 2, whereas in phase 1 the minor classes with fewer sample counts were considered for the analysis. It indicates that all the classifiers are performing well when their sample counts are high for the training process. The phase 1 accuracy can also be improved if the remaining 3 data sample counts are averaged using some data augmentation process.

The performance of the proposed GSFTNN model shows better accuracy on both phase operations. This is achieved because of its multi-head attention block. At the same time, the performance of its previous model RNN shows a lesser accuracy rate when compared to all the other models due to its sequential operation process. Also, the performances of the LSTM show a slighter improvement in its experiment by eradicating unwanted information from its neurons. The ResNet models are very efficient in general but their nature of having more layer count makes the model suffer from getting the optimum features for the analytic process. The training time attainments of the verified algorithms are shown in Fig. 9 where the performances of GSFTNN indicate a betterment due to the nature of the simultaneous operation. All the algorithms are showing a betterment in the phase 2 model where the sample counts are comparatively minimum than the phase 1 operation.

Zooming on to measure the effectiveness of the proposed algorithm, further comparative analysis of the four (4) deep learning algorithms was conducted. Figs. 10 and 11 illustrate the confusion matrix and the receiver operating characteristic (ROC), respectively.

A confusion matrix is a table that is used to evaluate the performance of a classifier by comparing the predicted classes with the true classes. It is a useful tool for understanding the strengths and weaknesses of a classifier, and it can be used to identify areas for improvement. The matrix is made up of four quadrants that represent the number of true positives, false positives, true negatives, and false negatives. These values can

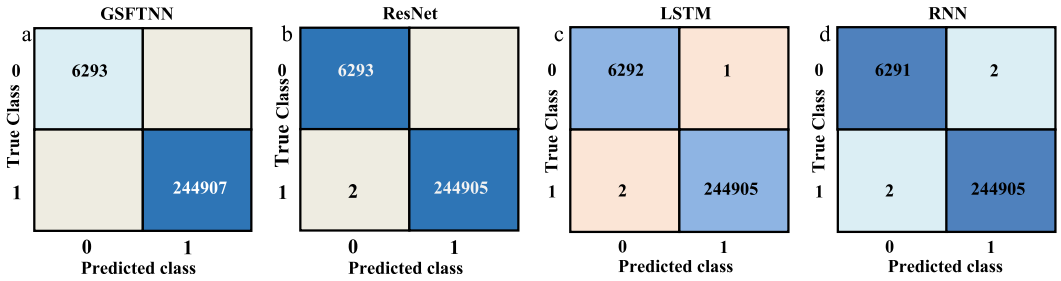


Fig. 10. (a) Confusion matrix of the GSFTNN; (b) Confusion matrix of the ResNet; (c) Confusion matrix of the LSTM; (d) Confusion matrix of the RNN.

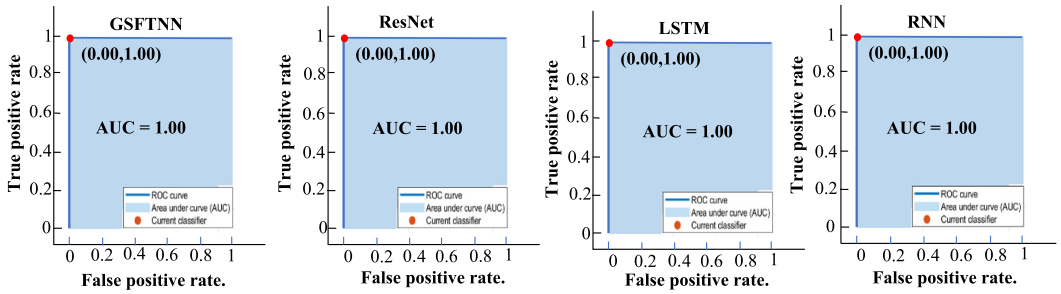


Fig. 11. (a) ROC of the GSFTNN; (b) ROC of the ResNet; (c) ROC of the LSTM; (d) ROC of the RNN.

Table 10
Comparison of existing algorithms.

Reference	Year	Dataset	Algorithm	Accuracy	Recall	F1 score	Precision
Altaha et al. (2020)	2020	Generated dataset	CNN	98.1			
		Generated dataset	FNN	98.8			
		Generated dataset	GRU	98.1			
		Generated dataset	LSTM	98.0			
		Generated dataset	RNN	98.0			
Chen, Dewi, Huang, and Caraka (2020)	2020	Bank marketing dataset	RF+SVM	89.0	91.37		97.91
		Bank marketing dataset	RF+KNN	88.6	90.8		96.91
		Bank marketing dataset	RF+RF	90.99	91.22		98.10
Khoei, Aissou, Hu, and Kaabouch (2021)	2021	CICDDoS-2019	KNN	94.6	94.4		
		CICDDoS-2019	RF	94.0	94.0		
		CICDDoS-2019	Stacking	96.0	97.3		
		CICDDoS-2019	Naïve Bayes	87.0	77.1		
Abdelkhalek and Govindarasu (2022)	2022	WUSTL-IIoT-2018	ANN	98.40	98.02	98.97	**99.57**
	2023	WUSTL-IIoT-2018	**GSFTNN**	**99.12**	**98.85**	**99.2**	99.26

then be used to calculate various performance metrics such as f1 score, recall, precision, and accuracy. A binary classifier system's performance as the discrimination threshold changes are graphically depicted by a ROC curve. The genuine positive rate (sensitivity) against the false positive rate (specificity) at various threshold settings is plotted on the ROC curve. By comparing a classifier algorithm's performance to that of a random guessing classifier, it is frequently possible to gauge how well it performs. The area under curve (AUC) is a frequently used performance statistic for classifiers. While a classifier that performs no better than random guessing has an AUC of 0.5, a perfect classifier has 1. ROC curves are frequently used to compare the effectiveness of various classifiers or the effectiveness of a single classifier in various scenarios (see Table 10).

5. Conclusion

Cybersecurity in the smart grid has become critically important on a multi-stakeholder scale and worldwide for academics and entrepreneurs. The danger to smart grid cyber security is significantly expanding in scope as energy systems gain pervasive intelligence and communications capabilities throughout their operational processes. Numerous SCADA networks have been the target of significant cyber-attacks that badly damaged the operational control circuits and related components. In other instances, a cyber-attacker creates a knockoff by imitating the distinctive algorithmic flow embedded into the SCADA network. The internet and wireless connectivity have made it possible for hackers to quickly achieve their objectives in several industries. As a result, we proposed a GSFTNN approach with a GSF feature selection model to develop a trustworthy deep learning algorithm. Extensive experiments were conducted using the WUSTL-IIOT-2018 ICS SCADA cyber security dataset. The experimental results reveal that the proposed GSFTNN algorithm surpasses RNN, LSTM, and ResNet in both accuracy and training time. The proposed algorithm's adeptness in categorizing data and predicting outcomes expeditiously serves as a testament to its robustness. The results provide empirical evidence that the GSFTNN algorithm is a

more efficacious and efficient algorithm than the aforementioned algorithms. Performance comparison of the proposed GSFTNN model to its latest counterparts' results as in Ahakonye et al. (2023) will be investigated in the future. In addition, we will also focus on the key factors such as spectral variability in SCADA systems that could influence the model's performance. We will further improve the model's generalization ability to unfamiliar scenarios.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data is available online.

References

Abdelkhalek, M., & Govindarasu, M. (2022). ML-based anomaly detection system for DER DNP3 communication in smart grid. In *Proc. 2022 IEEE int. conf. cyber secur. resilience, CSR 2022* (pp. 209–214). http://dx.doi.org/10.1109/CSR54599.2022.9850313.

Ahakonye, L. A. C., Nwakanma, C. I., Lee, J. M., & Kim, D.-S. (2023). Agnostic CH-DT technique for SCADA network high-dimensional data-aware intrusion detection system. *IEEE Internet of Things Journal*, 1. http://dx.doi.org/10.1109/jiot.2023.3237797.

Al Husaini, M. A. S., Habaebi, M. H., Hameed, S. A., Islam, M. R., & Gunawan, T. S. (2020). A systematic review of breast cancer detection using thermography and neural networks. *IEEE Access*, *8*, 208922–208937. http://dx.doi.org/10.1109/ACCESS.2020.3038817.

Altaha, M., Lee, J. M., Aslam, M., & Hong, S. (2020). Network intrusion detection based on deep neural networks for the SCADA system. *Journal of Physics: Conference Series*, *1585*(1), http://dx.doi.org/10.1088/1742-6596/1585/1/012038.

Altunay, H. C., Albayrak, Z., Ozalp, A. N., & Cakmak, M. (2021). Analysis of anomaly detection approaches performed through deep learning methods in SCADA systems. In *HORA 2021-3rd int. congr. human-computer interact. optim. robot. appl. proc.* http://dx.doi.org/10.1109/HORA52670.2021.9461273.

Avola, D., Cinque, L., Fagioli, A., & Foresti, G. L. (2022). SIRe-networks: Convolutional neural networks architectural extension for information preservation via skip/residual connections and interlaced auto-encoders. *Neural Networks*, *153*, 386–398. http://dx.doi.org/10.1016/j.neunet.2022.06.030.

Balla, A., Habaebi, M. H., Islam, M. R., & Mubarak, S. (2022). Applications of deep learning algorithms for supervisory control and data acquisition intrusion detection system. *Cleaner Engineering and Technology*, *9*(June), Article 100532. http://dx.doi.org/10.1016/j.clet.2022.100532.

Chen, J. I.-Z., & Chang, J.-T. (2020). Applying a 6-axis mechanical arm combine with computer vision to the research of object recognition in plane inspection. *Journal of Artificial Intelligence and Capsule Networks*, *2*(2), 77–99. http://dx.doi.org/10.36548/jaicn.2020.2.002.

Chen, R. C., Dewi, C., Huang, S. W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, *7*(1), http://dx.doi.org/10.1186/s40537-020-00327-4.

Cheng, L., Wu, X. H., & Wang, Y. (2018). Artificial flora (AF) optimization algorithm. *Applied Sciences*, *8*(3), http://dx.doi.org/10.3390/app8030329.

Cherifi, T., & Hamami, L. (2018). A practical implementation of unconditional security for the IEC 60780 — 5 — 101 SCADA protocol. *International Journal of Critical Infrastructure Protection*, *20*, 68–84. http://dx.doi.org/10.1016/j.ijcip.2017.12.001.

Gao, H., Zhang, Y., Chen, Z., Xu, S., Hong, D., & Zhang, B. (2023). A multi-depth and multi-branch network for hyperspectral target detection based on band selection. *IEEE Transactions on Geoscience and Remote Sensing*, *61*, 1. http://dx.doi.org/10.1109/tgrs.2023.3258061.

Hassan Malik, S. P., Alam, Muhammad Mahtab, Kuusik, Alar, & Moullec, Yannick Le (2020). Narrowband internet of things (NB-IoT) for industrial automation. In *Wirel. autom. as an enabler next ind. revolut* (pp. 65–87).

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. In *Proc. IEEE comput. soc. conf. comput. vis. pattern recognit., Vol. 2016-Decem* (pp. 770–778). http://dx.doi.org/10.1109/CVPR.2016.90.

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Identity mappings in deep residual networks. In *LNCS: vol. 9908, Lect. notes comput. sci. (including subser. lect. notes artif. intell. lect. notes bioinformatics)* (pp. 630–645). http://dx.doi.org/10.1007/978-3-319-46493-0_38.

Hoffmann Souza, M. L., da Costa, C. A., de Oliveira Ramos, G., & da Rosa Righi, R. (2021). A feature identification method to explain anomalies in condition monitoring. *Computers in Industry*, *133*, http://dx.doi.org/10.1016/j.compind.2021.103528.

Jasperneite, J., Sauter, T., & Wollschlaeger, M. (2020). Why we need automation models. *IEEE Industrial Electronics Magazine*, *14*(1), 29–40.

Jmila, M. I. K., & Houda (2022). Adversarial machine learning for network intrusion detection: A comparative study. *Computer Networks*, *214*(109073).

Karim, S. H., Fazle, Majumdar, Somshubra, & Darabi, Houshang (2019). Multivariate LSTM-FCNs for time series classification. *Neural Networks*, *116*, 237–245.

Khan, R. U., Zhang, X., Alazab, M., & Kumar, R. (2019). An improved convolutional neural network model for intrusion detection in networks. In *Proc. - 2019 cybersecurity cyberforensics conf. CCC 2019, No. Ccc* (pp. 74–77). http://dx.doi.org/10.1109/CCC.2019.000-6.

Khoei, T. T., Aissou, G., Hu, W. C., & Kaabouch, N. (2021). Ensemble learning methods for anomaly intrusion detection system in smart grid. In *IEEE int. conf. electro inf. technol., Vol. 2021-May* (pp. 129–135). http://dx.doi.org/10.1109/EIT51626.2021.9491891.

Kirubakaran, S. S. (2020). Study of security mechanisms to create a secure cloud in a virtual environment with the support of cloud service providers. *Journal of Trends in Computer Science and Smart Technology*, *2*(3), 148–154. http://dx.doi.org/10.36548/jtcsst.2020.3.004.

Kumar, D., & S, D. S. (2020). Enhancing security mechanisms for healthcare informatics using ubiquitous cloud. *Journal of Ubiquitous Computing and Communication Technologies*, *2*(1), 19–28. http://dx.doi.org/10.36548/jucct.2020.1.003.

Lee, J. M., & Hong, S. (2020). Keeping host sanity for security of the SCADA systems. *IEEE Access*, *8*, 62954–62968. http://dx.doi.org/10.1109/ACCESS.2020.2983179.

Liu, Q., & Wang, B. (2022). Neural extraction of multiscale essential structure for network dismantling. *Neural Networks*, *154*, 99–108. http://dx.doi.org/10.1016/j.neunet.2022.07.015.

Lopez Perez, R., Adamsky, F., Soua, R., & Engel, T. (2018). Machine learning for reliable network attack detection in SCADA systems. In *Proc. - 17th IEEE int. conf. trust. secur. priv. comput. commun. 12th IEEE int. conf. big data sci. eng. trust. 2018* (pp. 633–638). http://dx.doi.org/10.1109/TrustCom/BigDataSE.2018.00094.

Maglaras, L. A., & Jiang, J. (2014). Intrusion detection in SCADA systems using machine learning techniques. In *Proc. 2014 sci. inf. conf. SAI 2014* (pp. 626–631). http://dx.doi.org/10.1109/SAI.2014.6918252.

Mokhtari, S., Abbaspour, A., Yen, K. K., & Sargolzaei, A. (2021). A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electron*, *10*(4), 1–13. http://dx.doi.org/10.3390/electronics10040407.

Montalban, J. O. N., Iradier, E., & Member, G. S. (2020). NOMA-based 802. In *11n for industrial automation, Vol. 8*.

Ozdag, M. (2018). Adversarial attacks and defenses against deep neural networks: A survey. *Procedia Computer Science*, *140*, 152–161. http://dx.doi.org/10.1016/j.procs.2018.10.315.

P, A., Hong, J. C. D., Gao, L., Yao, J., & Zhang, B. (2020). Graph convolutional networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(9), 5966–5978. http://dx.doi.org/10.1109/TGRS.2020.3015157.

Pliatsios, D., Sarigiannidis, P., Lagkas, T., & Sarigiannidis, A. G. (2020). A survey on SCADA systems: Secure protocols, incidents, threats and tactics. *IEEE Communications Surveys and Tutorials*, *22*(3), 1942–1976. http://dx.doi.org/10.1109/COMST.2020.2987688.

Rousopoulou, V., et al. (2022). Cognitive analytics platform with AI solutions for anomaly detection. *Computers in Industry*, *134*, Article 103555. http://dx.doi.org/10.1016/j.compind.2021.103555.

Samdarshi, R., Sinha, N., & Tripathi, P. (2016). A triple layer intrusion detection system for SCADA security of electric utility. In *12th IEEE int. conf. electron. energy, environ. commun. comput. control (E3-C3), INDICON 2015* (pp. 1–5). http://dx.doi.org/10.1109/INDICON.2015.7443439.

Sarker, I. H. (2022). AI-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. *SN Computer Science*, *3*(2), 1–20. http://dx.doi.org/10.1007/s42979-022-01043-x.

Selvarajan, S., Shaik, M., Ameerjohn, S., & Kannan, S. (2020). Mining of intrusion attack in SCADA network using clustering and genetically seeded flora-based optimal classification algorithm. *IET Information Security*, *14*(1), 1–11. http://dx.doi.org/10.1049/iet-ifs.2019.0011.

Singh, V. K., Ebrahem, H., & Govindarasu, M. (2019). Security evaluation of two intrusion detection systems in smart grid SCADA environment. In *2018 north am. power symp. NAPS 2018*. http://dx.doi.org/10.1109/NAPS.2018.8600548.

Singh, P., Garg, S., Kumar, V., & Saquib, Z. (2015). A testbed for SCADA cyber security and intrusion detection. In *2015 int. conf. cyber secur. smart cities, ind. control syst. commun. SSIC 2015 - proc* (pp. 1–6). http://dx.doi.org/10.1109/SSIC.2015.7245683.

Smith, A., & Fressoli, M. (2021). Post-automation. *Futures*, *132*(June), Article 102778. http://dx.doi.org/10.1016/j.futures.2021.102778.

Teixeira, M. A., Salman, T., Zolanvari, M., Jain, R., Meskin, N., & Samaka, M. (2018). SCADA system testbed for cybersecurity research using machine learning approach. *Future Internet*, *10*(8), http://dx.doi.org/10.3390/fi10080076.

V, D. S. (2020). Automatic spotting of sceptical activity with visualization using elastic cluster for network traffic in educational campus. *Journal of Ubiquitous Computing and Communication Technologies*, *2*(2), 88–97. http://dx.doi.org/10.36548/jucct.2020.2.004.

Wang, W., Harrou, F., Bouyeddou, B., Senouci, S. M., & Sun, Y. (2022). A stacked deep learning approach to cyber-attacks detection in industrial systems: application to power system and gas pipeline systems. *Cluster Computing*, *25*(1), 561–578. http://dx.doi.org/10.1007/s10586-021-03426-w.

Wu, X., Hong, D., & Chanussot, J. (2022). Convolutional neural networks for multimodal remote sensing data classification. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, http://dx.doi.org/10.1109/TGRS.2021.3124913.

Wu, X., Hong, D., & Chanussot, J. (2023). UIU-net: U-net in U-net for infrared small object detection. *IEEE Transactions on Image Processing*, *32*, 364–376. http://dx.doi.org/10.1109/TIP.2022.3228497.

Yang, H. F., & Chen, Y. P. P. (2019). Representation learning with extreme learning machines and empirical mode decomposition for wind speed forecasting methods. *Artificial Intelligence*, *277*, Article 103176. http://dx.doi.org/10.1016/j.artint.2019.103176.

Yang, H., Cheng, L., & Chuah, M. C. (2019). Deep-learning-based network intrusion detection for SCADA systems. In *2019 IEEE conf. commun. netw. secur. CNS 2019*. http://dx.doi.org/10.1109/CNS.2019.8802785.

Yang, Y., McLaughlin, K., Sezer, S., Yuan, Y. B., & Huang, W. (2014). Stateful intrusion detection for IEC 60870 − 5 − 104 SCADA security. In *IEEE power energy soc. gen. meet., Vol. 2014-Octob* (pp. 5–9). http://dx.doi.org/10.1109/PESGM.2014.6939218, no. October.

Publication V

Risk Accessment of Machine Learning Algorithms on Manipulated Dataset in Power Systems

1st Sayawu Yakubu Diaba
School of Technology and Innovations
University of Vaasa
Vaasa, Finland
sdiaba@uwasa.fi

2nd Miadreza Shafie-khah
School of Technology and Innovations
University of Vaasa
Vaasa, Finland
miadreza.shafiekhah@uwasa.fi

3rd Mike Mekkanen
School of Technology and Innovations
University of Vaasa
Vaasa, Finland
mike.mekkanen@uwasa.fi

4th Tero Vartiainen
School of Technology and Innovations
University of Vaasa
Vaasa, Finland
Tero.Vartiainen@uwasa.fi

5th Mohammed Elmusrati
School of Technology and Innovations
University of Vaasa
Vaasa, Finland
moel@uwasa.fi

2023 International Conference on Future Energy Solutions (FES) | 979-8-3503-3230-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/FES57669.2023.10182751

***Abstract*—The emergence of the communication infrastructure in power systems has increased the variety and sophistication of network assaults. Intrusion Detection Systems' (IDS) importance has increased in relation to network security. IDS, however, is no longer secure when confronted with adversarial examples, and attackers can boost assault success rates by tricking the IDS. As a result, resilience must be increased. This paper assesses the Decision Tree, Logistic regression, Support Vector Machines (SVM), Naïve Bayes, K-Nearest Neighbours (KNN), and Ensemble's effectiveness. Using the WUSTL-IIoT-2021 dataset and CIC-IDS2017 dataset, we train the algorithms on the unmanipulated dataset and then train the algorithms on the manipulated dataset. Per the simulation results, the accuracy and prediction speed drop on the manipulated dataset while the training time rises.**

***Index Terms*—Communication infrastructure, Intrusion detection systems, Network security, Power systems**

I. Introduction

The growing interconnectedness of smart systems has led to increased concerns regarding their security vulnerabilities [1]. These systems are extensively utilized in several areas, including intelligent industries, smart cities, healthcare, and, many more, making security a critical issue [2]. Security measures such as authentication, authorization, encryption, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS), have been employed to mitigate these vulnerabilities. Despite these efforts, these systems remain vulnerable to cyber-attacks [3].

The energy sector is not an exception and thus, the sector faces a wide range of cyber-attacks, including physical, internal, external, and cyber threats. Cyber threats, in particular, can emerge from any location and pose significant challenges for the industry. Although it is impossible to eliminate these threats [4], mitigation strategies can help reduce their impact. However, implementing these strategies often requires significant financial resources and effort. In addition, cyber-attack mitigations can affect negatively and create psychological consequences that can be detrimental to the industry, leading to a decline in performance and potentially affecting national economies. Therefore, it is crucial to prioritize and implement effective cyber-attack mitigation strategies while also considering their potential economic and psychological impacts [5].

Utilizing the power of information technology, smart grid technology uses a two-way communication system to intelligently distribute energy [6]. This enables the integration of green technologies to satisfy environmental requirements. However, the use of communication technology also leaves the system open to many security risks. Even though many survey studies have addressed these problems and suggested solutions, most of them categorize attacks according to how they affect confidentiality, integrity, and availability [7]. As the prevalence of cyber-attacks continues to rise in the smart grid [8], the need for reliable and accurate IDS becomes increasingly critical [9].

The authors of [10] classified attack scenarios into control-based and measurement-based attacks. The control-based attack includes altering or fabricating control signals sent to the targeted power system assets. It can directly result in frequency and transient voltage instability, line overloading, load reduction, and cascading failures. The measurement-based attack seeks to compromise measurements in order to conceal or falsely represent the system's current state, impair observability, and ultimately deceive operators or control systems.

It's critical to comprehend the possible repercussions [11] of data manipulation on the accuracy of IDS algorithms. Any system that relies on these algorithms to detect malicious traffic may be at risk if the data has been manipulated. As such, it is crucial to identify and address the challenges associated with data manipulation in IDS algorithms to improve their reliability and effectiveness. In this paper, we aim to explore the impact of data manipulation on the performance of machine learning algorithms. We will evaluate the strengths

and weaknesses of various approaches used to mitigate these effects. By examining the limitations and vulnerabilities of these algorithms in the face of manipulated data, we can develop more robust and effective systems to safeguard against cyber-attacks [11].

The remaining sections of the paper are structured as follows. The system model is presented in Section II, and the data description is summarized in Section III. The experimental analysis is presented in Section IV of the paper, and the conclusions are presented in Section V.

II. System Model

The modeling approach used in this paper is borrowed from Linnartz et. al. as shown in Fig.1. The figure depicts a simplified schematic summary of the crucial components of the Cyber-Physical Systems (CPS), including the manipulation strategy. Information and Communication Technology (ICT) is used to link the power system's assets (physical system) to the central control system (cyber system).

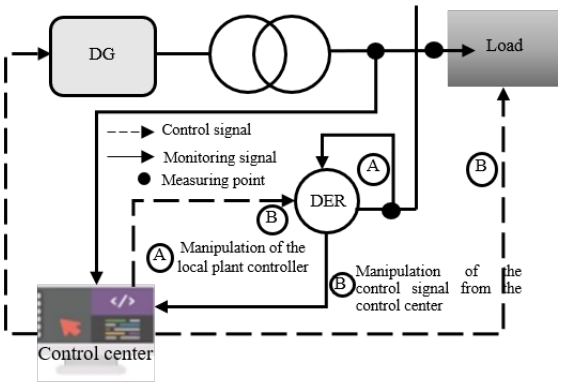


Fig. 1. Overview of the cyber-physical system. .

Supervisory control and data acquisition (SCADA) systems are used by Transmission System Operators (TSOs) and Distribution System Operators (DSOs) to continuously monitor and control the transmission and distribution assets of the power system (closed-loop control). Standardized protocols like IEC 61850, IEC 60870-5-104 or DNP3 are used for communication between the central management system and the assets. Typically, the information is transmitted through private channels, but neither of these protocols supports security features like authentication or integrity protection [12]. Thus, the communication channels are open to online threats.

For this experiment, the following assumptions are made.

- The attacker gains access and can interact with the assets according to the rules of the communication protocols. As a result, the attacker can alter control signals, create new ones, or stop all contact with the central control center.
- The attacker gained entry to the DSO's communication network. Due to earlier extensive reconnaissance operations, the attacker also has adequate knowledge of the control system, the power system, and its resources. He can then use the communication system to implement manipulation strategies by using denial of information attacks to transmit control signals to every connected DER.
- The attacker can attack and make data manipulations without the DSO noticing.

III. Data Description

Our proposed model's simulation evaluation stage utilizes the WUSTL-IIoT-2021 dataset [13] and the WUSTL-IIoT-2018 dataset. The WUSTL-IIoT-2021 dataset contains network data from an Industrial Internet of Things (IIoT) system. The dataset was developed using an IIoT testbed to support cybersecurity research, with the aim of emulating real-world industrial systems as accurately as possible. This testbed also allows for the simulation of actual cyber-attacks, providing a realistic environment for evaluating the effectiveness of the models. The normal and abnormal attacks are included in the WUSTL-IIOT-2018 dataset for ICS (SCADA) Cybersecurity research which closely matches true real-world data. It was constructed using a SCADA system testbed, which enabled the execution of an authentic cyber-attack [14].

A. Data pre-processing

The data preprocessing along with the data-cleaning process is depicted in Fig. 2. In the data pre-processing stage, columns with headings 'StartTime', 'LastTime', 'SrcAddr', 'DstAddr', 'sIpId', 'dIpId,' are removed from the data because they are specific to certain types of attacks. Including them in the training data would result in the algorithms being too specialized and not being able to accurately generalize to new, unseen data.

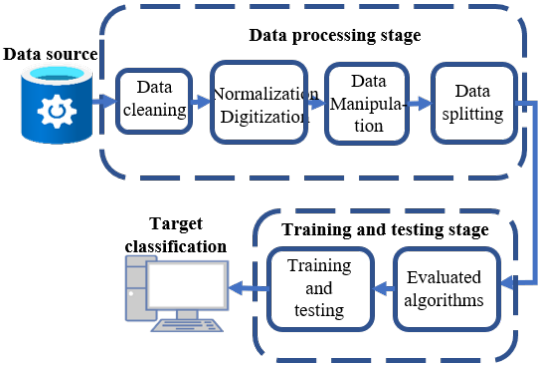


Fig. 2. The experimental model design.

Therefore, they were excluded to ensure the algorithms' ability to make accurate predictions on a wider range of input data. In the normalization and digitization stage, with respect to the target column, we represented normal with 0 and attack with 1. This column is manipulated by editing some 0's to 1

and some 1's to 0. The dataset is then split into a 70:30 ratio and used for the experiment.

However, Pearson's Correlation Coefficient (PCC) analysis is carried out on the dataset to perceive if it includes correlated features. This method was required because it guarantees the elimination of over-fitting. The PCC is mathematically given as

$$PCC = \frac{\sum (x_i + x_j)(y_i + y_j)}{\sum \sqrt{(x_i + x_j)^2 (y_i + y_j)^2}} \quad (1)$$

The symbol x_i represents the content of the variable in the dataset while x_j is referring to the average value of that variable. Similarly, y_i represents the values of the sample y_j represents the average value of that variable in the sample.

Correlation coefficients are statistical measures that describe the strength and direction of the relationship between two variables. A correlation matrix is typically square, with each variable listed on the table's rows and columns. The diagonal elements of the matrix are always 1 since each variable is perfectly correlated with itself, and the off-diagonal elements represent the correlation coefficients between the corresponding pairs of variables. A high positive correlation coefficient between two variables indicates that they tend to increase or decrease together, while a high negative correlation coefficient indicates that they tend to move in opposite directions. Correlation matrices score between -1 and +1 are often used to analyze the relationships between variables and identify data patterns and trends [15].

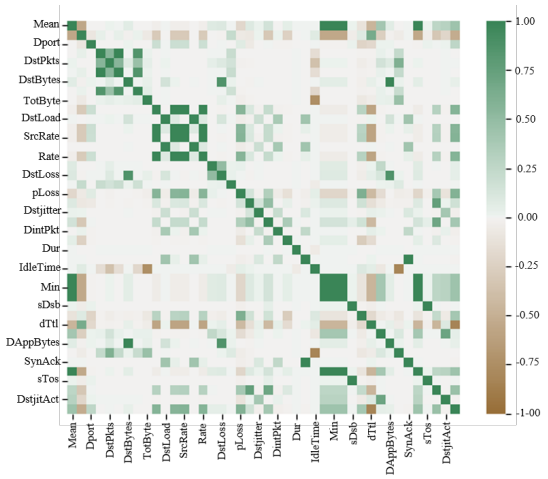


Fig. 3. The correlation matrix for the wustle-2021 dataset.

The effectiveness of the algorithm's categorization is assessed using a confusion matrix [16]. A confusion matrix is a method to evaluate how well a classifier algorithm is performing. It is composed of four fundamental parameters, including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [17].

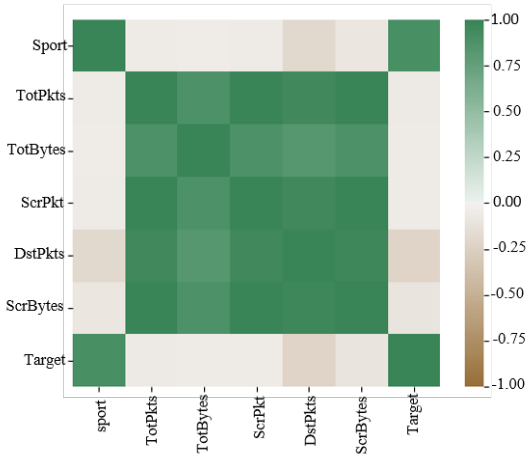


Fig. 4. The correlation matrix for the wustle-2018 dataset.

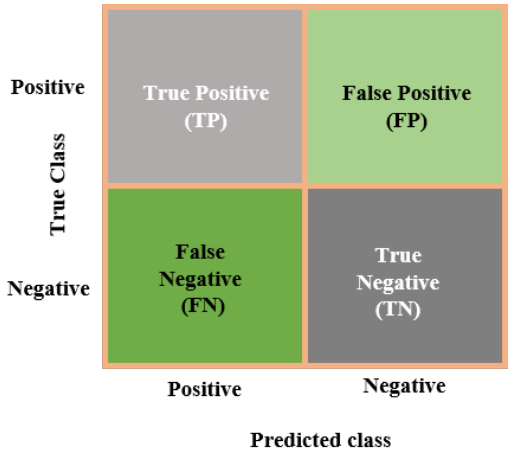


Fig. 5. The confusion matrix.

TP refers to when the algorithm correctly predicts a positive class, while TN indicates when the algorithm correctly predicts a negative class. FP happens when the algorithm predicts a positive class, but the actual class is negative, while FN refers to when the algorithm predicts a negative class, but the actual class is positive. Several performance metrics can be calculated based on these parameters, including accuracy, F1-score, precision, and recall. Accuracy measures the overall correctness of the algorithm's predictions. The algorithm's capability to correctly predict positive classes is measured by precision. The algorithm's capacity to recognize positive classes is measured by the recall, also known as sensitivity. Finally, the F1-score is a harmonic mean of precision and recall, providing a combined metric that balances both measures. The formulas are given as

follows [18], [19]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F - score = \frac{2(Presicion * Recall)}{Precision + Recall} \quad (5)$$

IV. Experimental Analysis

In this section, we will present the experimental analysis of the impact of data manipulation on the performance of IDS algorithms. We conducted a series of experiments to evaluate the accuracy, prediction speed, and training time of these algorithms when the data they analyze has been manipulated. We will present our findings and analyze the strengths and weaknesses of these algorithms in the context of manipulated data. Evaluating the performances of the algorithms, we selected the best-performing algorithm in each category. Performance is evaluated in terms of the likelihood of a successful detection [20]. The performance of these algorithms in terms of accuracy, considering the manipulated and unmanipulated WUSTL-IIoT-2018 dataset is presented in Fig. 6.

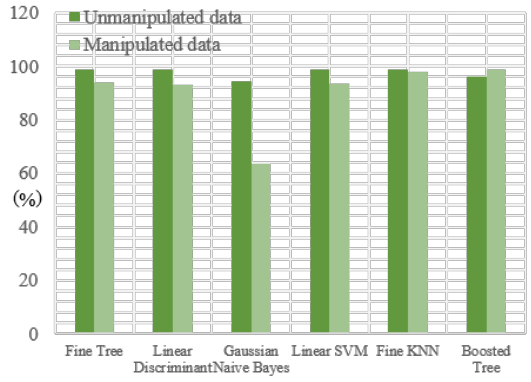


Fig. 6. Performance of the best-performing algorithms on the WUSTL-IIoT-2018 dataset.

The accuracy of the Fine Tree, Linear Discriminant, and Linear SVM all slightly declined with the manipulated data, as shown by the figure in Fig. 6. The Fine KNN showed little degradation while the Gaussian Naive Bayes experienced a significant decline. Surprisingly, the Booted tree's accuracy rose from 96.1% with the unmanipulated data, to 98% with the manipulated data.

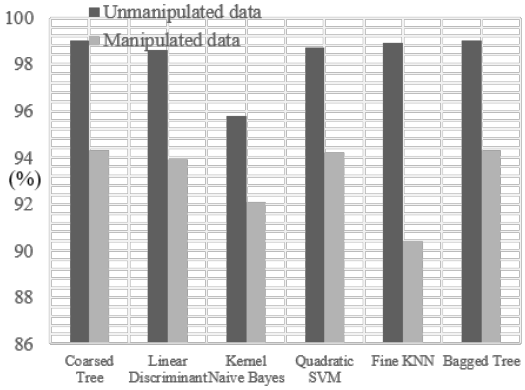


Fig. 7. Performance of the best-performing algorithms on the WUSTL-IIoT-2021 dataset.

The performance analysis of the best-performing algorithms is shown in Fig. 7. The Coarse Tree, Linear Discriminant, Quadratic SVM, and Bagged Tree classifiers experienced a moderate decrease in their performance when tested on manipulated data. On the other hand, the Fine KNN classifier showed a significant drop in accuracy, while the Kernel Naïve Bayes performed poorly on both unmanipulated data and manipulated data.

Algorithms	**Unmanipulated CIC-WUSTL-IIoT-2018**		**Manipulated CIC-WUSTL-IIoT-2018**	
	Prediction speed (obs/sec)	Training time (sec)	Prediction speed (obs/sec)	Training time (sec)
Fine Tree	920000	9.5195	72000	10.952
Linear Discriminant	260000	5.8716	530000	1.6454
Gaussian Naïve Bayes	140000	5.5346	390000	0.9483
Linear SVM	140000	89.756	73000	3266.1
Fine KNN	44000	111.6	49000	307.66
Booted Tree	120000	144.93	33000	340.99

Fig. 8. Prediction Speed and Training Time Comparison on the WUSTL-IIoT-2018 dataset.

The simulation results comparing prediction speed and training time on manipulated versus unmanipulated data are shown in Table 1. The Fine Tree, Linear SVM, and Booted Tree exhibited a decrease in prediction speed on the manipulated data, resulting in an increase in training time. Conversely, the Linear Discriminant and Gaussian Naïve Bayes classifiers demonstrated an increase in prediction speed accompanied by a reduction in training time on the manipulated data. The Fine KNN classifier, on the other hand, demonstrated an increase in both prediction speed and a decrease in training time on the manipulated data.

Algorithms	Unmanipulated CIC- WUSTL-IIoT-2021		Manipulated CIC- WUSTL-IIoT-2021	
	Prediction speed (obs/sec)	Training time (sec)	Prediction speed (obs/sec)	Training time (sec)
Coarse Tree	32000	2.2018	23000	3.6068
Linear Discriminant	120000	4.7506	190000	3.4319
Kernel Naïve Bayes	50	1998.8	41	10265
Quadratic SVM	58000	19.178	49000	81.972
Fine KNN	1100	127.64	1725	169.85
BaggedTree	30000	431.84	23000	431.84

Fig. 9. Prediction Speed and Training Time Comparison on the WUSTL-IIoT-2021 dataset.

The simulation results comparing prediction speed and training time on manipulated versus unmanipulated data (WUSTL-IIoT-2021) are presented in Table II. On the manipulated data, the Coarse Tree, Kernel Naïve Bayes, and Quadratic SVM exhibited a decrease in prediction speed, leading to an increase in training time. Conversely, the Linear Discriminant demonstrated an improvement in prediction speed and a decrease in training time on the manipulated data. In contrast, the Bagged Tree showed a reduction in prediction speed while maintaining a consistent training time across both datasets.

V. Conclusion

This paper exposes the potential risk of power systems to cyber-attacks where a malicious actor could alter the data. In intrusion detection systems, which are commonly used to defend networks against adversarial attacks, machine learning or deep learning techniques are extensively used due to their high accuracy and quick detection rates. The effectiveness of the top machine learning algorithms on manipulated datasets is examined in this paper. The experimental assessment made use of the WUSTL-IIoT-2021 dataset as well as the WUSTL-IIoT-2018 dataset. We evaluate the performance of several machine learning algorithms on both unmanipulated and manipulated datasets. According to experimental findings, except for the Boosted Tree algorithm, all algorithms' accuracy dropped, prediction speed decreased, and training time increased.

References

[1] K. Kumar, and V. Bhatnagar, "Machine Learning Algorithms Performance Evaluation for Intrusion Detection," Journal of Information Technology Management 13, no. 1 (2021): 42-61.

[2] X. Fu, N. Zhou, L. Jiao, H. Li, and J. Zhang, "The robust deep learning–based schemes for intrusion detection in the internet of things environments," Annals of Telecommunications 76, no. 5-6 (2021): 273-285.

[3] E. Degirmenci, I. Ozcelik, and A. Yazici, "Effects of Un targeted Adversarial Attacks on Deep Learning Methods," In 2022 15th International Conference on Information Security and Cryptography (ISCTURKEY), pp. 8-12. IEEE, 2022.

[4] J. A. Abraham, and V. R. Bindu, "Intrusion Detection and Prevention in Networks Using Machine Learning and Deep Learning Approaches: A Review," In 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), pp. 1-4. IEEE, 2021.

[5] S. K. Venkatachary, J. Prasad, and R. Samikannu, "Economic impacts of cyber security in energy sector: A review." International Journal of Energy Economics and Policy 7, no. 5 (2017): 250.

[6] P. A. Oyewole and D. Jayaweera, "Power System Security With Cyber-Physical Power System Operation," in IEEE Access, vol. 8, pp. 179970-179982, 2020, doi: 10.1109/ACCESS.2020.3028222.

[7] D. G. Kyrollos, K. Greenwood, J. Harrold, and J. R. Green, "Detection of false alarms in the NICU using pressure sensitive mat.," In 2021 IEEE Sensors Applications Symposium (SAS), pp. 1-5. IEEE, 2021.

[8] B. Li, B. Zhang, and D. S. Kirschen, "Cyber-Physical Attack Leveraging Subsynchronous Resonance." arXiv preprint arXiv:2207.04149 (2022).

[9] V. D. A. Kumar, "An Effective Comparative Analysis of Data Preprocessing Techniques in Network Intrusion Detection System Using Deep Neural Networks," Smart Intelligent Computing and Communication Technology 38 (2021): 14.

[10] H. He, and Jun Yan, "Cyber-physical attacks and defences in the smart grid: a survey," IET Cyber-Physical Systems: Theory and Applications 1, no. 1 (2016): 13-27.

[11] H. Wang, J. Ruan, B.Zhou, C. Li, Q. Wu, M. Q. Raza, and G. Cao, "Dynamic data injection attack detection of cyber physical power systems with uncertainties," IEEE Transactions on Industrial informatics 15, no. 10 (2019): 5505-5518.

[12] P. Linnartz, A. Winkens, and A. Ulbig, "Assessing the impact of cyber attacks manipulating distributed energy resources on power system operation," arXiv preprint arXiv:2207.07968 (2022).

[13] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of Industrial Internet of Things," in IEEE Internet of Things Journal 6 (2019), pp. 6822-6834.

[14] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, M. Samaka, "SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach," Future Internet 2018, 10, 76,

[15] L. A. C. Ahakonye, C. I. Nwakanma, J. -M. Lee and D. -S. Kim, "Efficient Classification of Enciphered SCADA Network Traffic in Smart Factory Using Decision Tree Algorithm," in IEEE Access, vol. 9, pp. 154892-154901, 2021, doi: 10.1109/ACCESS.2021.3127560.

[16] Y. Chen, C. Fan, and K. Chang, "Manufacturing intelligence for reducing false alarm of defect classification by integrating similarity matching approach in CMOS image sensor manufacturing," Computers and Industrial Engineering 99 (2016): 465-473.

[17] Q. Zhang, X. Chen, Z. Fang, and S. Xia. "False arrhythmia alarm reduction in the intensive care unit using data fusion and machine learning," In 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 232-235. IEEE, 2016.

[18] M. Qian, J. Luo, Y. Ge, C. Sun, X. Ge, and W. Huang, "Semantic-based false alarm detection approach via machine learning," In 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 60-66. IEEE, 2021.

[19] B. Taji, A. D. C Chan, and S. Shirmohammadi, "False alarm reduction in atrial fibrillation detection using deep belief networks," IEEE Transactions on Instrumentation and Measurement 67, no. 5 (2017): 1124-1131.

[20] T. Diskin, U. Okun, and A. Wiesel, "Learning to Detect with Constant False Alarm Rate," In 2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC), pp. 1-5. IEEE, 2022.