IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 January 2025, accepted 10 February 2025, date of publication 14 February 2025, date of current version 20 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3542227

RESEARCH ARTICLE

Machine Learning Techniques for Enhanced Intrusion Detection in IoT Security

HANADI HAKAMI[1], MUHAMMAD FAHEEM[2,3], AND MAJID BASHIR AHMAD[4]

[1]Department of Software Engineering, College of Engineering, University of Business and Technology, Jeddah 21361, Saudi Arabia
[2]School of Technology and Innovations, University of Vaasa, 65200 Vaasa, Finland
[3]VTT Technical Research Center of Finland Ltd., 02150 Espoo, Finland
[4]Department of Computer Sciences, COMSATS University Islamabad, Vehari 61000, Pakistan

Corresponding author: Muhammad Faheem (muhammfa@uwasa.fi)

The work of Muhammad Faheem was supported by the VTT Technical Research Center of Finland Ltd., Espoo, Finland.

ABSTRACT Network Intrusion Detection Systems (NIDSs) are fundamental to safeguarding computer networks. Intrusion detection systems must become more effective as new attacks are developed and networks grow. Anomaly-based automated detection stands out due to its superior performance among the various detection techniques. However, with the increasing complexity and frequency of cyberattacks, managing vast amounts of data remains challenging for anomaly-based NIDS. Therefore, it is necessary to find an efficient method for solving the problem by using classification with an intrusion detection system which analyzes enormous amounts of traffic data. This research introduces a new model that leverages machine learning (ML) and deep learning (DL) to enhance detection effectiveness and ensure reliability. The approach optimizes data preprocessing by integrating SMOTE for effective data balancing and Pearson's Correlation Coefficient (PCC) for feature selection. We compared several ML and DL techniques to detect and address the most efficient one for our pipeline. Compared with other approaches, LSTM and RF show superior results when tested on the WSN-DS, UNSW-NB15, and CIC-IDS 2017 datasets. Additionally, the proposed solution prevents biases from arising by addressing imbalanced datasets.

INDEX TERMS Intrusion detection, IoT, classification, machine/deep learning, random forests, long-short-term-memory.

I. INTRODUCTION

The fast-paced furtherance of advanced technologies, including big data, the Internet of Things (IoT), and cloud computing, coupled with our growing dependence on networked services in daily communication, has made networked computing indispensable. This, in turn, has heightened the importance of ensuring securing networks. Any potential weakness or danger can impact the entire network [1]. Conventional security methods, including firewalls and encryption technologies, face challenges as attackers develop sophisticated attacks. Addressing these challenges to safeguard our networked systems effectively [2] is crucial. Furthermore, cybersecurity researchers emphasize creating effective network intrusion detection systems (IDSs) to ensure secure networks. IDSs aim to provide the availability, protection of data privacy and accuracy during transmission between interconnected computers. They prevent unlawful access to a network and safeguard IT and communication systems within the network [3]. A network intrusion detection (NID) system observes computer systems and analyzes network data to detect potential unauthorized entries into the system [4]. Beyond detecting intrusions and monitoring network activity for potentially concerning or nefarious activities, it also identifies policy breaches. This makes it simpler for network administrators to stay vigilant against current threats [5]. In addition to accurately detecting known and unknown threats, these systems are designed to reduce false alarm rates. [2].

A NIDS consists of two techniques: malpractice detection and abnormality detection. Detection of misuse, also called pattern-based detection, relies on a pre-defined set of known

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

attacks and threats for detection [6]. The detection rate of this model is high, and there are few false alarms. Nevertheless, as networks and services expand, attackers continuously devise new, unidentified attacks, rendering the model susceptible to these threats [7]. For robust network security, a NIDS must be efficient and smart In recognizing and resisting familiar and unfamiliar attacks, like anomaly identification [6]. It can detect both known and unknown attacks, even though it has a high false alarm rate.

Implementing artificial intelligence (AI) enables systems and devices to gain knowledge from datasets with minimal manual intervention, and NID systems have successfully leveraged this functionality. ML and DL, both aspects of AI, played integral roles in creating a robust NID system. In an ML system, network traffic categorization and surveillance rely on manually extracted features. In contrast, a DL system, employing neural networks, can autonomously derive attributes from the dataset and subsequently carry out classification and detection [1]. A robust NID System can be deployed at the edge nodes of IoT networks to address potential cybersecurity concerns. M/DL-based NID systems have protected IoT networks for decades. It is, however, unclear whether they are effective. It is crucial to increase intrusion detection accuracy so that false alarms can be lessened and detection rates can be augmented. Recent attempts to improve performance have utilized multi-layer perceptrons, support vector machines, and other techniques. However, these methods demonstrate limitations and inefficiencies when applied to extensive datasets like system and network data. It is also noteworthy that the comparative study of M/Dl has never been done before.

To tackle these challenges, the intrusion detection system must analyze vast volumes of traffic data. The key lies in employing a classification technique that handles such extensive datasets. This research unveils a unique technique for identifying intrusions by performing a comparative analysis across three diverse datasets: WSN-DS, UNSW-NB15, and CIC-IDS 2017. Unlike previous research focused on single datasets, this multi-dataset analysis comprehensively evaluates intrusion detection methods in varied network environments and attack scenarios. Moreover, this research fills a notable void in the current body of research by being the first to conduct a comprehensive contrastive analysis of M/DL techniques for detecting intrusions. Specifically, it evaluates the performance of Random Forests (RF) and Long Short-Term Memory networks (LSTM) across different datasets. By comparing these algorithms, this work sheds light on their relative performance and effectiveness, enhancing the generalizability of the findings and progressing the area of intrusion detection research with insights applicable across a wide range of practical environments. Moreover, this study aims to strengthen the performance and reliability of NIDS in real-world environments. With the growing complexity of cyberattacks, efficiently analyzing large network traffic is critical for security. The proposed model compares ML and DL techniques, offering a scalable solution for NID. Using SMOTE and PCC addresses the challenge of imbalanced datasets. This research provides practical guidance for developing more effective NIDS and selecting the best algorithms for various network environments. Furthermore, the proposed study pioneers a feature filtering method using Pearson's Correlation Coefficient (PCC) for intrusion detection models, setting a higher correlation threshold of 0.97, surpassing the conventional 0.95 used in prior studies. This precision-driven approach significantly boosts accuracy, achieving up to a 3% improvement in model performance compared to lower threshold models. By prioritizing highly correlated features, our methodology optimizes intrusion detection systems for real-world scenarios, ensuring superior reliability and effectiveness in identifying intrusion activities.

This research offers the following major contributions:

- This study presents a novel framework for an NID system tailored to IoT environments using M/DL methodologies. Unlike prior studies that focused on a single dataset, it introduces a comparative analysis of three diverse datasets: WSN-DS, UNSW-NB15, and CIC-IDS 2017. This multi-dataset evaluation thoroughly assesses intrusion detection techniques across varied network environments and attack scenarios.
- A new feature filtering technique is proposed, utilizing a higher correlation threshold of 0.97, improving upon the conventional threshold of 0.95 used in earlier research. This method effectively removes redundant feature correlations, reducing training complexity. It also enhances feature selection accuracy, leading to up to a 3% improvement in model performance compared to models using lower thresholds.
- The study bridges a critical gap in the literature by offering the first comprehensive comparative analysis of M/DL methods for intrusion detection. By evaluating RF and LSTM across multiple datasets, the research highlights their strengths and weaknesses. This broader analysis enhances the generalizability of the findings and provides actionable insights for designing intrusion detection systems in real-world applications.
- The proposed approach addresses the challenge of dataset imbalance in intrusion detection, ensuring greater reliability in identifying intrusion activities. By focusing on correlated features and mitigating dataset biases, the approach optimizes the performance and robustness of intrusion detection systems, making them more effective for real-world deployment.

Detailed below is the framework for the remainder of the study. Section II reviews IDS. It is described in Section III how different M/D Learning techniques are applied to the proposed model of NID systems. There are sections on evaluation methodology and investigation of the results in Section IV. The ultimate Section V covers findings and outlines the next research directions.

II. LITERATURE REVIEW

Cybersecurity experts are working on a model to spot documented and undiscovered network attacks and stop them from harming the network. Since Denning's [8] pioneering work, numerous promising solutions have emerged to these cybersecurity challenges. The keywords for the literature review of NID encompass essential terms within the scope of NID, including ''Network intrusion detection,'' ''Machine learning,'' ''Deep learning,'' ''LSTM,'' ''Random Forest,'' ''Comparative study,'' ''Evaluation metrics,'' ''Datasets,'' and ''Cybersecurity.'' The acceptance-rejection criteria guide the selection process, focusing on concentrates that address M/DL techniques for NID while ensuring relevance to cybersecurity, methodological rigor, clear presentation of results and performance metrics, and publication in peer-reviewed journals or reputable conference proceedings. To compile a comprehensive review, literature is collected from well-regarded platforms such as IEEE Xplore and the ACM Digital Library, SpringerLink, ScienceDirect, PubMed, and Google Scholar, guaranteeing the availability of diverse scholarly works. These criteria and databases collectively facilitate a thorough examination of pertinent research, fostering a nuanced understanding of advancements and challenges in network intrusion detection methodologies. As we'll see shortly, the methods created for IDSs (IDS) fall into two categories: ML and DL.

ML has played and continues to play a crucial role in IDSs. ML algorithms include supervised learning methods like Decision Trees, SVM, and Naïve Bayesian analysis and unsupervised learning algorithms methods like K-means clustering and Self Organized Map [2]. These algorithms mainly aim to improve a system's ability to detect threats. Training data is utilized to identify attacks and potential risks. ML algorithms are commonly applied to address regression, classification, and clustering problems. Earlier research heavily relied on datasets like NSL-KDD, DARPA, and KDD-CUP99. Despite certain models showing acceptable outcomes, these data collections are outdated and feature only basic attack types [1], [2]. Training an IDS (IDS) in the ongoing, ever-enlarging network necessitates a substantial dataset. Utilizing established ML models designed for small datasets won't lead to an effective model [2].

DL is a type of ML that works with multi-layered artificial neural networks [2]. This method learns from unlabeled or unorganized data, unlike other methods [9]. DL effectively does certain things, like building IDSs (IDS). It's strong because the algorithms are reliable, scalable, and can handle various kinds of data [2]. DL algorithms were originally made to tackle tricky problems like pattern recognition, search engines, and machine translation [10].

Mahbooba et al. [11] contributed to the field of eXplainable Artificial Intelligence (XAI) by exploring its application in cybersecurity, specifically within IDS. Addressing the challenge of uninterpretable ML models in cybersecurity, the research employed transparent decision tree models to boost trust management in IDS. Using simple decision tree models, their study aimed to provide interpretable insights into model behavior, facilitating comprehension by human experts. Through experimentation on a widely recognized dataset, their paper evaluated decision tree-based approaches and compared their performance with literature algorithms. Ultimately, this research advances the understanding and implementation of XAI principles in cybersecurity, focusing on improving trust management in IDS. Mankodiya et al. [12] applied ML algorithms, notably stacking approaches for trust management in autonomous vehicles, and have shown promising results, achieving high accuracies like 98.44%. The VeRiMi dataset was analyzed using a decision tree-based random forest method, and results showed impressive 98.43 and 98.5% accuracy metrics.

Lin et al. [13] improved anomaly detection in networks by introducing a system that observes changes over time, employing LSTM and the Attention Mechanism for enhanced effectiveness. They ensured accuracy using the SMOTE algorithm, achieving a checking accuracy of about 96.2%. Hai and Nam [15], Meliboev et al. [14], and Ravinder Reddy et al. [18] compared RNN, LSTM, and GRU models, with Hai and Nam [15] finding similar performances around 93.9% accuracy on CSE-CIC-IDS2018 and CIC-IDS2017 datasets. Debicha et al. [21] focused on enhancing system resilience against intentional confusion, strategically placing an adversarial detector that outperformed an individual detector for spotting evasion attacks through transfer learning.

According to preceding research, Al-Zewairi et al. [16] used an RNN for binary categorization, achieving an impressive 98.99% accuracy on the UNSW-NB15 benchmark data. Assis et al. [17] suggested a GRU-based model for safeguarding SDN against intrusion attacks through direct flow inspection. The model was evaluated on the CICDDoS2019 and CSE-CIC-IDS2018 datasets, attaining recall, accuracy, and F1-scores 94.7%, 97.1%, and 97% for the respective datasets, with the latter dataset showing these values.''

Hosseininoorbin et al. [19] investigated the utilization of Google's Edge TPU for an efficient IoT-edge Network IDS. They evaluated expanded DNN frameworks for processing performance, power consumption and data traffic classification performance compared to a power-efficient embedded CPU. Khan [20] developed a CRNNIDS incorporating CNN for spatial attributes and RNN for time-related understanding, demonstrating its effectiveness with an impressive accuracy of 97.75% on the CSE-CIC-IDS2018 dataset. Other research exploring combined methods that combine both CNN and RNN are presented in the exploration of Cao et al. [22], Wang et al. [23], and Zhang et al. [24]. Bastola et al. [25] integrated LSTM and GRU models for intrusion detection, exhibiting superior performance compared to independent LSTM and GRU models. However, Zhao et al. [26] raised concerns about potential insufficient feature capture due to the distinct architectures within this combined model. Aldwairi et al. [27] presented a filter that distinguishes between criteria for filtering and incorporating

TABLE 1. Literature review summary.

Reference	Proposed Approach	Limitations
Lin et al. [13], Meliboev et al. [14]	Introduced a system for anomaly detection in networks using LSTM and the Attention Mechanism, achieving 96.2% checking accuracy with the SMOTE algorithm.	Restricted to certain datasets and may not perform effectively in other contexts.
Hai and Nam et al. [15]	Compared RNN, LSTM, and GRU models, achieving around 93.9% accuracy on various datasets.	Lack of exploration of novel architectures or techniques.
Al-Zewairi et al. [16]	Achieved 98.99% accuracy on the UNSW-NB15 benchmark dataset using an RNN for binary classification.	Lack of evaluation on other datasets.
Assis et al. [17], Ravinder Reddy et al. [18]	Proposed a GRU-based model for safeguarding SDN against intrusion attacks, achieving recall, accuracy, and F1-scores of 94.7%, 97.1%, and 97%, respectively, on the CICDDoS2019 and CSE-CIC-IDS2018 datasets.	Limited evaluation on datasets beyond CICDDoS2019 and CSE-CIC-IDS2018.
Seyedehfaezeh Hosseini-noorbin et al. [19]	Examined the application of Google's Edge TPU for optimizing an IoT-edge Network IDS, evaluating scaled DNN models for processing performance, power consumption, and traffic classification effectiveness in comparison to a low-energy embedded CPU.	Limited to the assessment of scaled deep neural network models.
Khan [20]	Created a CRNNIDS that integrates CNN for capturing local features and RNN for analyzing temporal patterns, showcasing its efficacy with an accuracy of 97.75% on the CSE-CIC-IDS2018 dataset.	Lack of evaluation on datasets beyond CSE-CIC-IDS2018.

in NID. This approach employed n-grams derived from the prefixes of Snort signatures to separate legitimate traffic, thereby decreasing computational requirements and improving detection speed.

Fu et al. [28] illustrated a case of this technique by integrating CNN with Bi-LSTM networks. Their method was assessed with the NSL-KDD dataset, resulting in an impressive F1-score of 90.7% and an accuracy of 89.65%. Kanna and Santhi [29] introduced an enhanced technique that accounts for both spatial and temporal dimensions. They utilized Optimized CNN (OCNN) and a unique LSTM termed Hierarchical Multi-scale LSTM (HMLSTM). Evaluations on three datasets (NSL-KDD, UNSW-NB15, and ISCX-IDS-2012) yielded F1 scores of 91.47%, 97.61%, and 98.13%, respectively. In a subsequent study, Kanna and Santhi [30] developed a model called BWO-CONV-LSTM, which merged CNN and LSTM, using Black Widow Optimized (BWO) for hyperparameter tuning. Wang et al. [31] proposed RUIDS, a resilient unsupervised IDS leveraging transformer-based self-supervised learning. RUIDS demonstrated significant AUC improvements across four datasets, including a 9.04% increase on UNSW-NB15 and a 9.58% rise on CICIDS-WED, showcasing its ability to handle diverse anomaly interference with minimal impact on performance. Sharma et al. [32] introduced a DL model for NID, evaluated on the UNSW-NB15 dataset. The model attained 84% performance and 91% accuracy with a dataset that had balanced class distributions. Xiong et al. [33] presented AIDTF, which employed adversarial training utilizing an a-model, d-model, and t-module, attaining enhanced accuracy compared to a range of assaults using adversarial samples during training, demonstrating effectiveness against both established and novel attacks, exceeding other techniques.

M/DL-based NID systems have proven highly effective in detecting IoT network attacks in recent years. However, the assessment methods reveal constraints and prove ineffective when applied to extensive datasets, such as system and network data. Notably, a comprehensive evaluation of M/DL methods for IDS has yet to be undertaken. To address the challenges faced by M/DL-based NID systems in effectively assessing extensive datasets and the lack of comprehensive comparative studies, a multifaceted approach is necessary. There is a need to develop advanced assessment methodologies specifically tailored to handle large-scale datasets. Secondly, fostering collaborative research efforts within the cybersecurity community is essential to conducting comprehensive comparative studies across diverse datasets. By standardizing evaluation metrics and datasets, researchers can facilitate meaningful comparisons of the effectiveness of various M/DL methodologies. To achieve this, the suggested method improves detection performance. It lowers false positives by utilizing RF and LSTM networks with a filter-based technique employing PCC and trained on diverse and extensive datasets: CIC-IDS 2017, UNSW-NB15, and WSN-DS. Additionally, it contrasts the outcomes of RF (ML) and LSTM (DL). These techniques have demonstrated effectiveness in addressing classification challenges. Moreover,

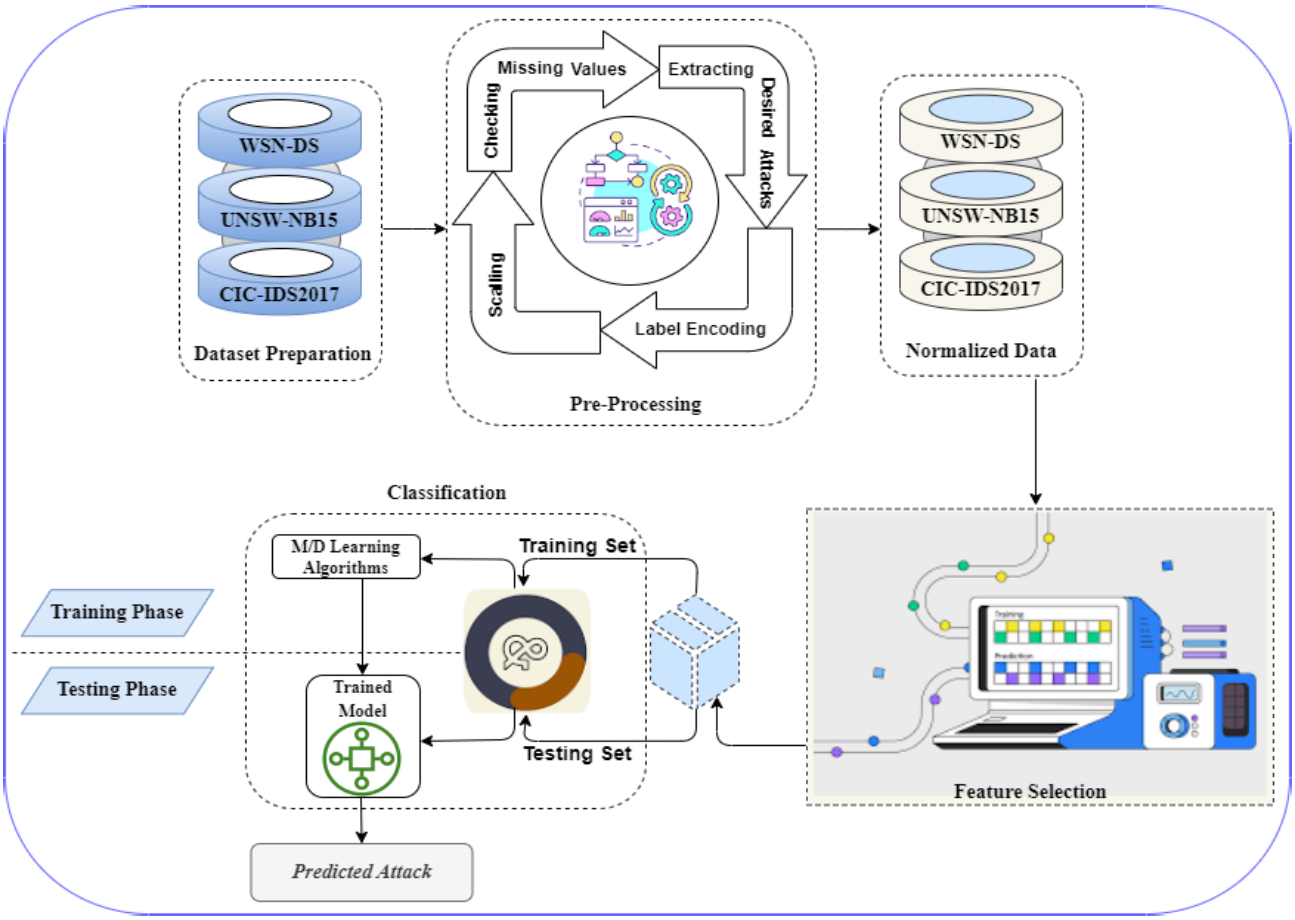


FIGURE 1. Proposed approach.

by addressing the challenge of imbalanced datasets, these models mitigate biases and ensure reliable performance across various classes of network traffic. The review of existing research summary is presented in Table 1.

III. PROPOSED APPROACH

A. OVERVIEW

In this section, IoT networks are exposed to vulnerabilities and attacks. Figure 1 depicts the IDS model presented in this study. The method encompasses four main phases: *Data Preparation* to access data; *Data Preprocessing* to standardize data; *Feature Selection* to determine optimal features; and the *Classification*. Each step in the proposed system holds significance and contributes significantly to its overall performance. The dominant objective of this study is to ascertain or launch the efficacy of different IDS methods, specifically RF and LSTM, and to compare them. The workflow is depicted in Figure 1. However, the specifics are as follows:

B. DATASET PREPARATION

The initial phase in creating a dependable IDS involves choosing a suitable dataset that encompasses both normal and malicious records and reflects real-world scenarios. This investigation employs newly available CIC-IDS2017, UNSW-NB15, and WSN-DS datasets. These datasets provide fresh and unique normal and malicious traffic data, minimizing redundancy and ensuring relevance to contemporary environments.

1) WSN-DS DATASET

WSN-DS is designed to identify both normal and malicious traffic by surveilling node numbers in wireless networks equipped using sensors. The dataset captures data via the LEACH routing protocol, depicted by 18 attributes and 1 label. It comprises standard 374661 records along with four categories of Denial-of-Service (DoS) attacks: flooding, Grayhole, blackhole, and TDMA [34] as resentedrep in Table 2.

Statistics outlining the packet quantities for each category in the WSN-DS dataset.

2) UNSW-NB15 DATASET

The UNSW-NB15 dataset includes one standard class and 9 specific attack categories: Exploits, Worms, Reconnaissance, Generic, DoS, Fuzzers, Analysis, Backdoors, and Shellcode. Established by the Australian Centre for Cyber Security (ACCS), the data was compiled from three

TABLE 2. Statistics outlining the packet quantities for each category in the WSN-DS dataset.

Attack Type	No. of Records
Normal	340,066
Grayhole	14,596
Blackhole	10,049
TDMA	6,638
Flooding	3,312
Total	374,661

TABLE 3. Statistics outlining the packet quantities for each category in the UNSW-NB15 dataset.

Attack Type		No. of Records	
Normal	93000	Reconnaissance	13987
Generic	58871	Analysis	2677
Exploits	44525	Backdoor	2329
Fuzzers	24246	Shellcode	1511
DoS	16353	Worms	174
Total Packets		**257673**	

TABLE 4. Statistics outlining the packet quantities for each category in the CIC-IDS2017 dataset.

Attack Type	No. of Records
DDoS	128,027
BENIGN	97,718
Total	225,745

practical sources: BID (Symantec Corporation), CVE (Common Vulnerabilities and Exposures), and MSD (Microsoft Corporation) (Microsoft Security Bulletin) [35]. It features 42 attributes and a single indicator specifying if an entry is classified as regular traffic or an attack. The dataset contains 257,673 records, with 93,000 labeled normal and 164,673 representing numerous attack varieties. Detailed statistics on the packet count for each attack category are detailed in Table 3.

3) CIC-IDS2017 DATASET

The CIC-IDS2017 dataset includes 11 innovative attacks, including Brute Force, PortScan, and DoS, web-based attacks like XSS and SQL Injection, and FTP-Patator and SSH-Patator. Formulated by the Canadian organization for Cybersecurity, this collection provides diverse attack scenarios for analysis and research purposes. This dataset employs 78 features and 1 label having 225745 records with 128,027 entries labeled as DDoS and 97,718 entries spanning BENIGN as outlined in Table 4 [35].

C. PRE-PROCESSING

The WSN-DS, UNSW-NB15, and CIC-IDS2017 datasets are initially obtained from *Kaggle* and scrutinized for missing or duplicated entries. Extraneous network frames are removed from the UNSW-NB15 dataset, and instances from various categories are compiled. Categorical data is converted into numerical parameters utilizing a specific label encoding approach, while quantitative data is normalized using min-max scaling. Each dataset, saved as a.csv file, is imported into Google Colab, including 12 gigabytes (GB) of memory and a Tesla K80 GPU.

In WSN-DS, 19 features are available, consisting of 18 numerical values and 1 attack category indicating whether it is classified as normal or an attack category as shown in Figure 2. For UNSW-NB15, there are 44 features, including 39 quantitative figures, 4 descriptive variables, and 1 indicator specifying whether it is a standard or an intrusion category. Unnecessary columns for classification are eliminated, plus the assessment targets six categories: Normal, Generic, Exploits, Fuzzers, DoS, and Reconnaissance, as shown in Figure 3. The decision to exclude specific classes with very few instances is a deliberate option to increase the model's durability. The strategic removal of highly imbalanced classes aims to balance handling class imbalances and ensure the model's broader applicability. While removing imbalanced classes may seem counterintuitive, this action is implemented to manage the overall dataset inequality and allow the model to interpret the data to learn efficiently. Removing classes with highly skewed sample distributions avoids the model becoming excessively biased towards the dominant class, enabling it to concentrate on extracting significant patterns from the data. The CIC-IDS2017 dataset includes 79 attributes, consisting of 78 numerical values and 1 attack type label, which indicates whether the entry is BENIGN or DDoS, as illustrated in Figure 4.

Categorical attributes are adapted into quantative values via label transformation. In particular, the categorical features such as state, proto, service, and attack-cat for the UNSW-NB15 dataset, Attack type for the WSN-DS dataset, and Label for the CIC-IDS2017 dataset are subjected to a label encoding process, converting their initial values into integers. The datasets undergo min-max normalization following the normalization procedure described in Equation 1. This ensures that numerical attributes with different ranges are standardized to a consistent scale.

$$X_{\text{new}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

In Equation 1, *Xnew* signifies the newly normalized value, where *Xmin* and *Xmax* depict the lowest and highest values within the X column, respectively. The full set of measurements in the X column, extending from *Xmin* to X_{max}, is scaled down to the normalized interval between 0 and 1. This scaling is essential for reducing significant fluctuations that might restrict convergence and maintains that every aspect is on a uniform scale, thus aiding in refinement. The resulting dataset is properly aligned wit standardized values following these pre-processing steps.

D. FEATURE SELECTION

Choosing particular features contributes to conserving storage space and lowering computational costs [36]. The

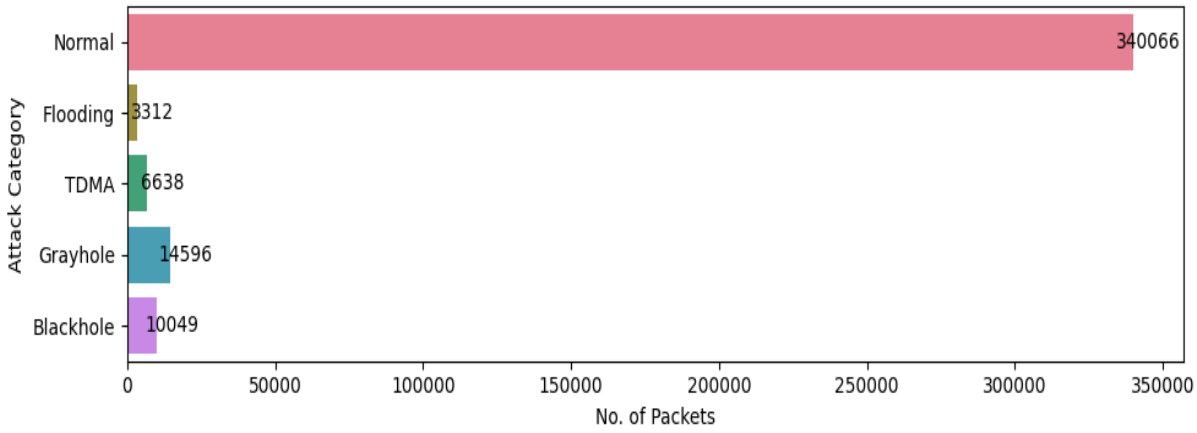


FIGURE 2. WSN-DS dataset.

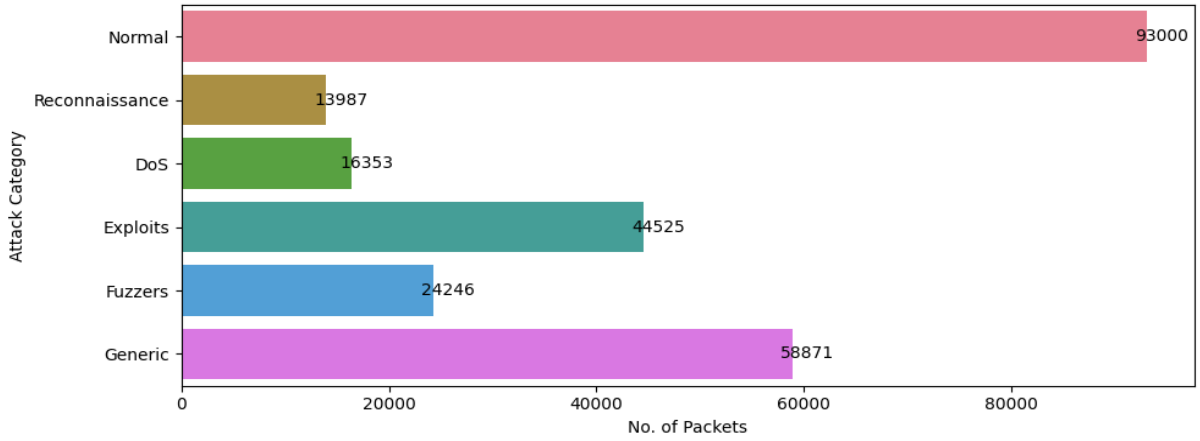


FIGURE 3. UNSW-NB15 dataset.

methods for feature selection can be categorized into the following groups:

- **Filter Techniques:** This approach uses correlation metrics to assess the association between features. Features are selected based on statistical values, with the choice made by comparing these values against a set threshold. Popular methods incorporate the Chi-square examination, relational analysis, and informational benefit [32].
- **Wrapper Techniques:** This tactic involves selecting a subset of features and training an ML model using them. The collection of properties is adjusted by incorporating or eliminating features depending on the model's performance. Popular techniques to select and eliminate forward and backward elimination [32].
- **Embedded Techniques:** This approach joins the benefits of incorporating methods for filtering and wrapping methods to identify the most effective features while preserving computational efficiency. Ordinary examples include the RF approach [37] and LASSO standardization [38].

In M/D learning, attributes are essential for optimizing data capacity optimization, reducing computational expenses, and enhancing overall model effectiveness. The suggested model Utilizes the faster and less computationally intensive filter method, specifically utilizing PCC, to select attributes pulled consider carefully the dataset.

- **Pearson's Correlation Coefficient (PCC):** The PCC, denoted as ρ, measures the association involving two probabilistic variables, represented as X and Y. It is mathematically expressed as:

$$\rho = \frac{\text{cov}(X, Y)}{\sigma(X)\sigma(Y)} \tag{2}$$

In this context, cov and σ symbolize the covariance and standard deviation, while p indicates the PCC value within the range of -1 to 1. When the uncertain quantities X and Y strongly associate, the PCC (ρ) approaches values near -1 or 1. In contrast, when X

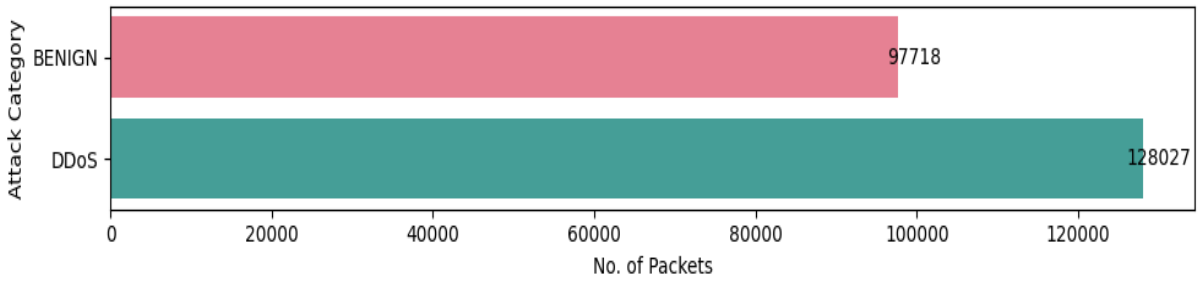


FIGURE 4. CIC-IDS 2017 dataset.

and Y lack correlation altogether, ρ equals 0. As a result, attributes with a high correlation are deemed redundant, and one can be eliminated. By appraising the associations amidst variables in the dataset, those with extremely high or low values are recognized as strongly associated. An affiliation matrix visually represents these associations, removing attributes that exceed a particular cutoff. The sample of the correlation matrix on the WSN-DS characteristics is shown in Figure. 5. The circles in Figure. 5 represent the strength of correlations between feature pairs in the dataset. Larger circles indicate stronger correlations, while smaller circles represent weaker ones. The colour of the circles and grids conveys the direction and intensity of the correlation: shades of blue typically signify negative correlations (closer to -1), where one variable increments as the other decreases, while shades of red correlations that are positive (closer to +1), where both variables increase or decrease together. Neutral colours, i.e., white, suggest little to no correlation (close to 0). This visual representation helps in quickly identifying strong associations between features.

In developing the model, features exhibiting a high correlation surpassing a threshold of 0.97 are sustained, while the rest are discarded. The threshold of 0.97 is chosen as it enhances accuracy by up to 3% compared to a threshold of 0.95 or lower. Specifically, in the WSN-DS dataset, the feature 'who CH' is excluded. After this removal, the updated dataset includes 17 features and 1 label. In the UNSW-NB15 dataset, the features 'ct_ftp_cmd', 'ct_srv_dst', 'dbytes', 'dloss', 'dwin', 'sbytes', and 'sloss' are discarded. After eliminating these columns, the revised dataset contains 35 features and 1 label. For CIC-IDS2017, the features 'Active Min', 'Average Packet Size', 'Avg Bwd Segment Size', 'Avg Fwd Segment Size', 'Bwd Header Length', 'Bwd Packet Length Std', 'ECE Flag Count', 'Flow IAT Max', 'Fwd Header Length.1', 'Fwd IAT Max', 'Fwd IAT Std', 'Fwd IAT Total', 'Fwd Packet Length Std', 'Fwd Packets/s', 'Idle Max', 'Packet Length Std', 'SYN Flag Count', 'Subflow Bwd Bytes', 'Subflow Bwd Packets', 'Subflow Fwd Bytes', 'Subflow Fwd Packets', and 'Total Length of Bwd Packets' are excluded. After removing these columns, the revised dataset comprises 56 features and 1 label.

E. CLASSIFICATION

The fundamental role of an NID system lies in categorizing activities as either normal or intrusive, which is carried out by an intrusive analysis engine. The proposed study focuses on utilizing two specific classifiers, RF and LSTM, as intrusive analysis engines due to their established effectiveness in handling classification problems. A consistent hyperparameter tuning strategy is adopted to ensure fair comparisons between models trained on the WSN-DS, UNSW-NB15, and CIC-IDS2017 datasets. Grid and random search techniques are utilized within the same predefined search space across all datasets. Key hyperparameters such as learning rate, number of epochs, batch size, and regularization parameters are tuned uniformly for all models. Furthermore, k-fold cross-validation is employed during the tuning process to minimize the impact of data splits and ensure robust parameter selection. The best-performing hyperparameter configurations for each model-dataset combination are recorded and provided in the supplementary materials for reproducibility. This approach aimed to ensure consistency across all models and datasets and was pivotal in achieving the best performance for each model. Each classification approach is explained in detail below.

1) RANDOM FORESTS (RF)

RF is a robust ensemble classifier widely used to analyze intrusion detection data and address classification and regression tasks. Its methodology involves constructing multiple decision trees (DT) during training, denoted as $T_1, T_2, \ldots, T_K$, and determining class labels based on the most prevalent outcome [39]. DTs divide the feature space into parts and label instances based on the predominant class within each segment. Every decision tree T_i can be depicted as a recursive binary structure, where individual node n signifies a feature test, and each leaf node denotes a class identifier. Let X represent the feature space and Y the set of class labels. Each decision tree T_i strives to learn a mapping function as follows:

$$f_i : X \rightarrow Y \tag{3}$$

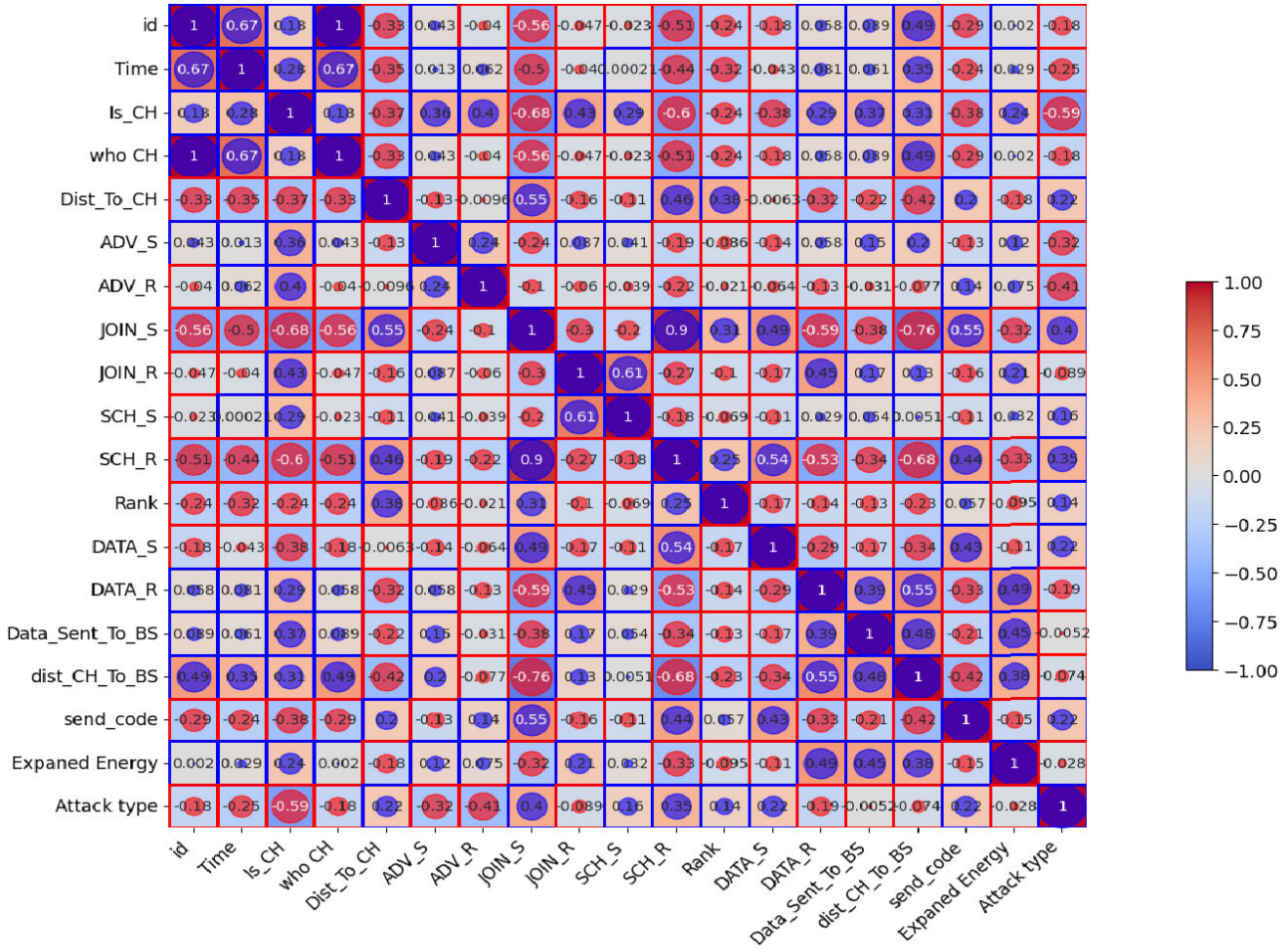


FIGURE 5. Sample of correlation matrix on WNSW-NB15.

As illustrated in Figure 6, the Random Forest algorithm handles a pre-processed batch of n instances, creating n unique decision trees utilizing different subsets of features. Each tree makes an independent classification, and the conclusive outcome is determined through a majority vote. The label assigned is the one with the most votes. Historical classification results demonstrate RF's effectiveness in handling such data, often surpassing other classifiers. The Random Forest algorithm offers additional benefits, such as enhanced accuracy over Adaboost and a minimized likelihood of overfitting.

RF is an ensemble technique that builds numerous DT throughout the training phase. When provided with a training dataset:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{4}$$

where x_i denotes the attribute vector and y_i indicates the class label, Random Forest constructs K decision trees utilizing different subsets of features. Each tree T_i makes an independent classification, and the overall decision is reached by consolidating the predictions from all trees through a majority voting process. The class assigned to a given sample has the highest occurrence among the predictions made by all decision trees. Let $C_i(x)$ represent the class label predicted by the i-th decision tree regarding the sample x. The ultimate prediction for x is obtained by:

$$\hat{y}(x) = \text{argmax}_y \sum_{i=1}^{K} I(C_i(x) = y) \tag{5}$$

where $\hat{y}(x)$ is the final predicted class identifier for x, $I(\cdot)$ is the indicator function, and K represents the complete count of decision trees.

2) LONG SHORT-TERM MEMORY (LSTM)

The fundamental idea behind the strength of LSTM lies in its potential to interpret and store parameters through storage cells across different time points. These memory cells undergo processing by gates, the activation functions of which are illustrated by gates. Figure 6 illustrates that the LSTM architecture consists of four gates: forgetting, updating, tanh activation, and output. Among these

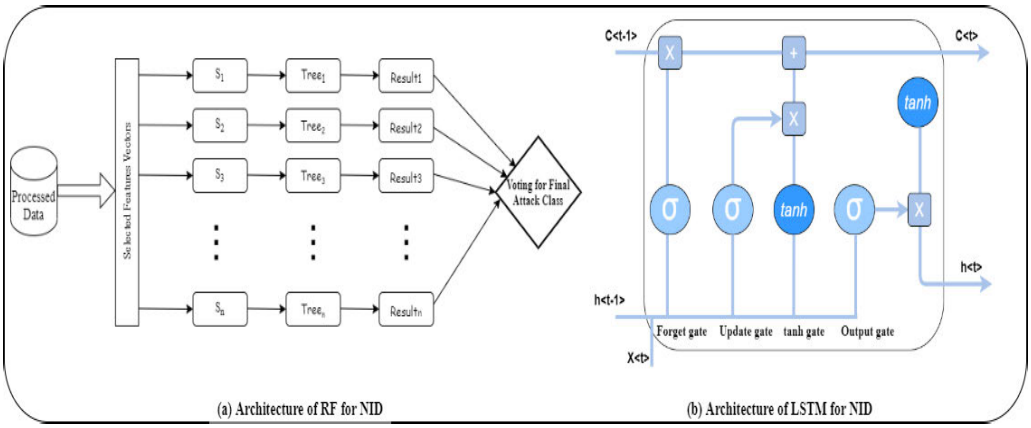


FIGURE 6. Architecture of RF and LSTM for NID.

TABLE 5. Classification performance metrics.

Metric	Formula
Precision (P)	$TP/(TP+FP)$
Recall (R)	$TP/(TP+FN)$
F1 Score (F)	$2PR/(P+R)$
Accuracy (A)	$(TP+TN)/(TP+TN+FP+FN)$

architectures, the training procedure is facilitated by modifying both the weights and the activation values functions. This adjustment aims to enable the effective generation of temporal relationships between the input and output data [40], [41].

LSTM networks manifest input and output values as $X(t)$ vectors of equal size. Information is retained or discarded in the forget gate based on $X(t)$ with $X(t-1)$. Additionally, the output is formulated using the sigmoid function and then scaled by the preceding cell state $C(t-1)$. The update component considers the input gate responsible for identifying the pertinent details for incorporation into generating $C(t)$. This generation process relies on the sigmoid and tanh functions facilitated through the tanh gate. The combined effect of these gates is combined with the product of the forget gate and $C(t-1)$, which ultimately produces $C(t)$. The resulting cell state $C(t)$ is then passed utilizing the tanh activation function, and this result is in conjunction with the output derived from the output gate's sigmoid activation. This sequence of operations yields the current hidden state $h(t)$, which signifies the result from the LSTM network. The equation below illustrates how the output is formulated:

$$O(t) = \sigma(b + U \times X(t) + W \times h(t-1)) \tag{6}$$

3) TRAINING AND TESTING

The dataset is partitioned into training and testing sets, with 80% allocated to training and 20% to testing, ensuring a balanced division. Each subset contains unique labels that denote either normal or attack categories. This split ensures enough data (80%) for the model to grasp patterns and relationships, while the remaining 20% acts as an independent set to evaluate how well the model generalizes. This ratio is widely accepted, finding an optimal solution between efficient training and accurate evaluation. A model's performance on new data is more realistically gauged by testing it on unseen data to prevent overfitting. The training set facilitates the training of the RF and LSTM models, whereas the testing subset is applied to assess their effectiveness. The correctness of the trained model is then evaluated with the testing data.

IV. EVALUATION

The system's performance is assessed using standard datasets: WSN-DS, UNSW-NB15, and CIC-IDS 2017 explained in Section III-B.

A. RESEARCH QUESTIONS

This research delves into the upcoming research questions:

- What are the comparative capability characteristics of RF and LSTM models in NID systems?
- What impact does the accuracy of the recommended approach in NID in comparison with the SOTA approach?
- What is the comparative performance of RF and LSTM models alongside other M/D Learning models?
- How does the filter influence the training performance of RF and LSTM models on a given dataset?
- How do RF and LSTM models perform on a balanced dataset?
- How does the proposed model perform under various noise types?

B. METRICS

Metrics for evaluation comprise accuracy, precision, recall, and F1 score, detailed in Table 5 [42]. Accuracy determines the fraction of correct predictions out of the total predictions

TABLE 6. Evaluation parameters of RF and LSTM on distinct datasets (In %).

	RF				LSTM			
Dataset	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
WSN-DS	99.71	99.71	99.71	99.71	99.41	99.43	99.41	99.41
UNSW_NB15	90.17	90.14	90.17	90.14	82.58	82.44	82.58	80.86
CIC-IDS 2017	99.99	99.99	99.99	99.99	99.96	99.95	99.95	99.94

TABLE 7. Model performance for different datasets using the optimal LSTM configuration.

Model	WSN-DS				UNSW-NB15				CIC-IDS 2017			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
1 layer, 64 units, 10 timesteps	99.41	99.43	99.41	99.41	82.58	82.44	82.58	80.86	99.96	99.95	99.95	99.94
2 layers, 128 units, 20 timesteps	**99.45**	**99.47**	**99.43**	**99.45**	**82.65**	**82.55**	**82.61**	**80.95**	**99.97**	**99.96**	**99.96**	**99.95**
3 layers, 256 units, 50 timesteps	99.43	99.46	99.42	99.44	82.60	82.49	82.62	80.89	99.97	99.96	99.96	99.95

TABLE 8. Model performance for different datasets using the optimal RF configuration.

Model	WSN-DS				UNSW-NB15				CIC-IDS 2017			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
50 estimators, max depth None	**99.7**	**99.71**	**99.71**	**99.7**	**90.17**	**90.14**	**90.17**	**90.14**	**99.99**	**99.99**	**99.99**	**99.99**
100 estimators, max depth None	99.68	99.69	99.68	99.68	90.12	90.09	90.12	90.10	99.98	99.98	99.98	99.98
200 estimators, max depth 20	99.72	99.73	99.72	99.72	90.15	90.11	90.15	90.13	99.99	99.99	99.99	99.99

made. Precision represents the true positives (TP) ratio to the combined number of TP and false positives (FP). Recall is the ratio of TP to the sum of TP and false negatives (FN). The F1 score is calculated as the harmonic mean of precision and recall. Higher F1 scores, approaching 1, reflect superior model performance. Table 5 lists the specific values for accuracy, recall, precision, and F1 score.

Several key metrics are essential in evaluating a classification model's performance. A **True Positive** (TP) arises when the model reliably forecasts the positive class, while a **True Negative** (TN) happens when it correctly determines as the negative class. On the other hand, a **False Positive** (FP) occurs when the model wrongly classifies the positive class. At the same time, a **False Negative** (FN) happens when it incorrectly identifies the negative class. These metrics facilitate assessing the model's correctness and reliability.

C. RESULTS AND DISCUSSION

1) COMPARATIVE ANALYSIS OF RF AND LSTM

A comparative analysis of the evaluation outcomes for the NID task using RF and LSTM models across various datasets is conducted to address the initial research question. Performance metrics for RF and LSTM on the WSN-DS, UNSW_NB15, and CIC-IDS 2017 datasets are summarized in Table 6. According to Table 6, RF consistently delivers high performance assessed using accuracy, precision, recall, and F1 score across all datasets. Specifically, RF attains an accuracy of 99.71%, 90.17%, and 99.99% on WSN-DS, UNSW_NB15, and CIC-IDS 2017 datasets, respectively. Alternatively, LSTM demonstrates strong performance with high accuracy, precision, recall, and F1 score across all datasets. On the WSN-DS dataset, LSTM has an accuracy of 99.41% and 99.96% on CIC-IDS 2017. However, LSTM's accuracy on UNSW_NB15 is slightly lower at 82.58%.

Moreover, to identify the optimal configuration of the LSTM and RF model, this study conducted a series of experiments by systematically varying key hyperparameters. The hyperparameter settings and their respective ranges and results are summarized in Table 7 and Table 8. The analysis of these experiments highlights that the hyperparameter setting (2 layers, 128 hidden units, and a sequence length of 20) yields the best results because it strikes a balance between model complexity and its capacity to learn meaningful temporal patterns from the data. This configuration has been adopted for subsequent experiments and model evaluation. The analysis of the experiments using RF models indicates that the hyperparameter setting of 50 estimators with no maximum depth yields the best results across all datasets. This configuration achieves near-perfect performance, with accuracy, precision, recall, and F1 scores consistently high, especially for the CIC-IDS 2017 dataset, which reaches an accuracy of 99.99%. This setting strikes an optimal balance between model complexity and generalization, making it suitable for further experiments and model evaluation.

For each configuration, the performance of the model is assessed using specific evaluation metric, e.g., accuracy, precision, recall, and F1-score. The results of these experiments, presented in Table 6, reveal the impact of varying hyperparameters on the model's effectiveness.

Moreover, results at different thresholds are mentioned in Figure 7. The proposed feature filtering technique with a

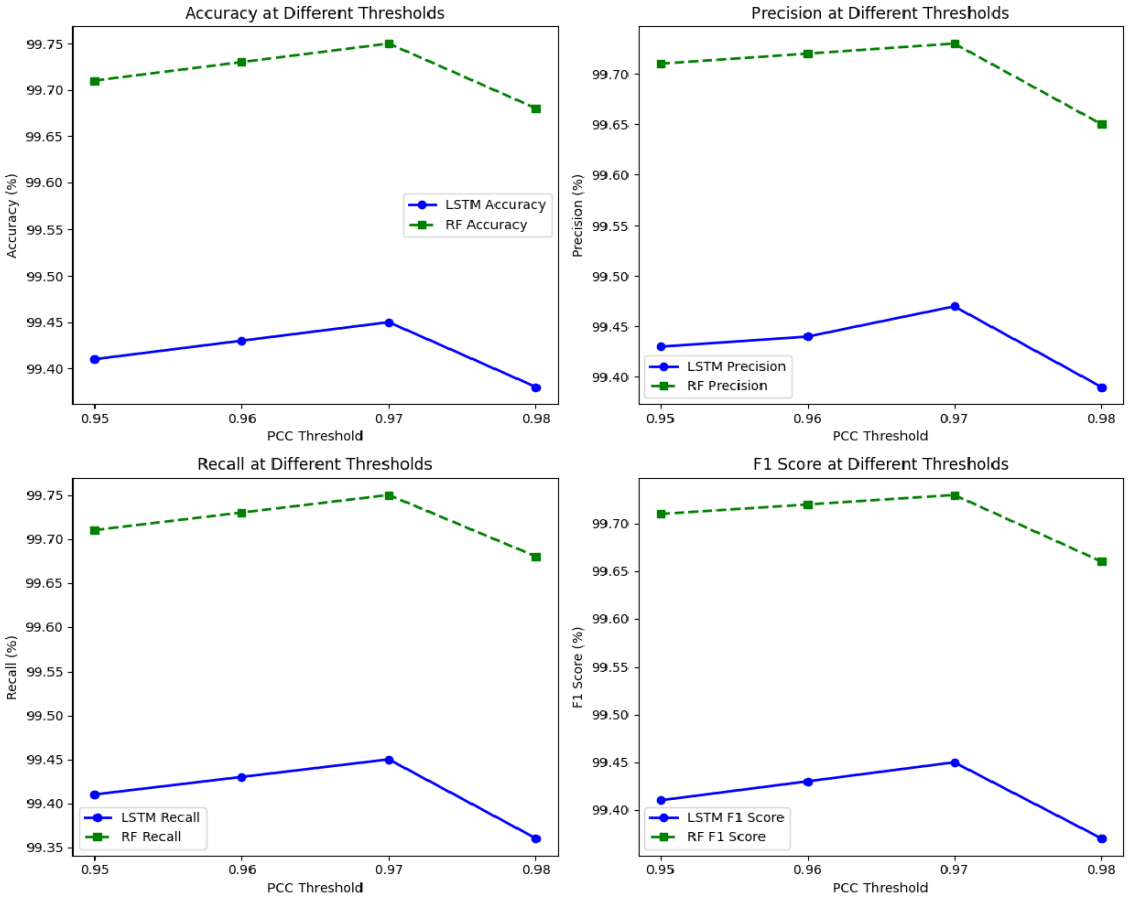


FIGURE 7. Performance comparison of LSTM and RF models at different PCC thresholds.

0.97 correlation threshold improves upon the conventional 0.95 threshold by more effectively removing redundant features, reducing training complexity, and enhancing feature selection accuracy. This results in a more efficient and streamlined model, leading to up to a 3% performance improvement compared to models using lower thresholds. Using a threshold of 0.98 can remove valuable features that, despite being somewhat correlated, still contribute unique predictive power. This may lead to an overly simplified model with reduced diversity in the feature set, potentially affecting its performance and generalization ability. The model could become biased, missing important data patterns.

The confusion matrix and ROC (Receiver Operating Characteristic) metrics are detailed in Figures 8 and 9. High precision is observed with fewer false positives, while high recall is noted with fewer false negatives. A high F1 score indicates the effective classification of each class in the test data. Table 6 shows that RF surpasses LSTM concerning overall accuracy, precision, recall, and F1 score. RF demonstrates enhanced performance in multiclass (WSN-DS, UNSW_NB15) and binary classification (CIC-IDS 2017) compared to LSTM. RF's effectiveness in managing tabular data, robustness against noise and irrelevant features, interpretability, transparency, quicker training times, and capability to handle imbalanced datasets contribute to its superior performance over LSTM. The analysis concludes that the RF model is more effective for NID than the LSTM model.

Despite the advanced and powerful nature of DL for feature extraction, as discussed in Section I, feature importance in RF models enhances their interpretability metrics, which aid in pinpointing essential features for NID. Their capability is evident across datasets of varying sizes, and they are generally less resource-intensive than deep learning models, which often struggle with large data requirements. By employing an ensemble strategy, RF boost performance and minimizes the chances of overfitting and surpasses LSTM by integrating multiple weak learners into a strong model. Additionally, RF models are a practical choice, especially when resources are constrained, due to their capacity to cope with non-linear relationships without significant feature engineering, resilience to anomalies, and efficiency in dealing with erroneous or unrelated data. Moreover, to clarify the computational efficiency of the proposed methodology, we have calculated the training and inference times for RF and LSTM models independently mentioned in Section IV.

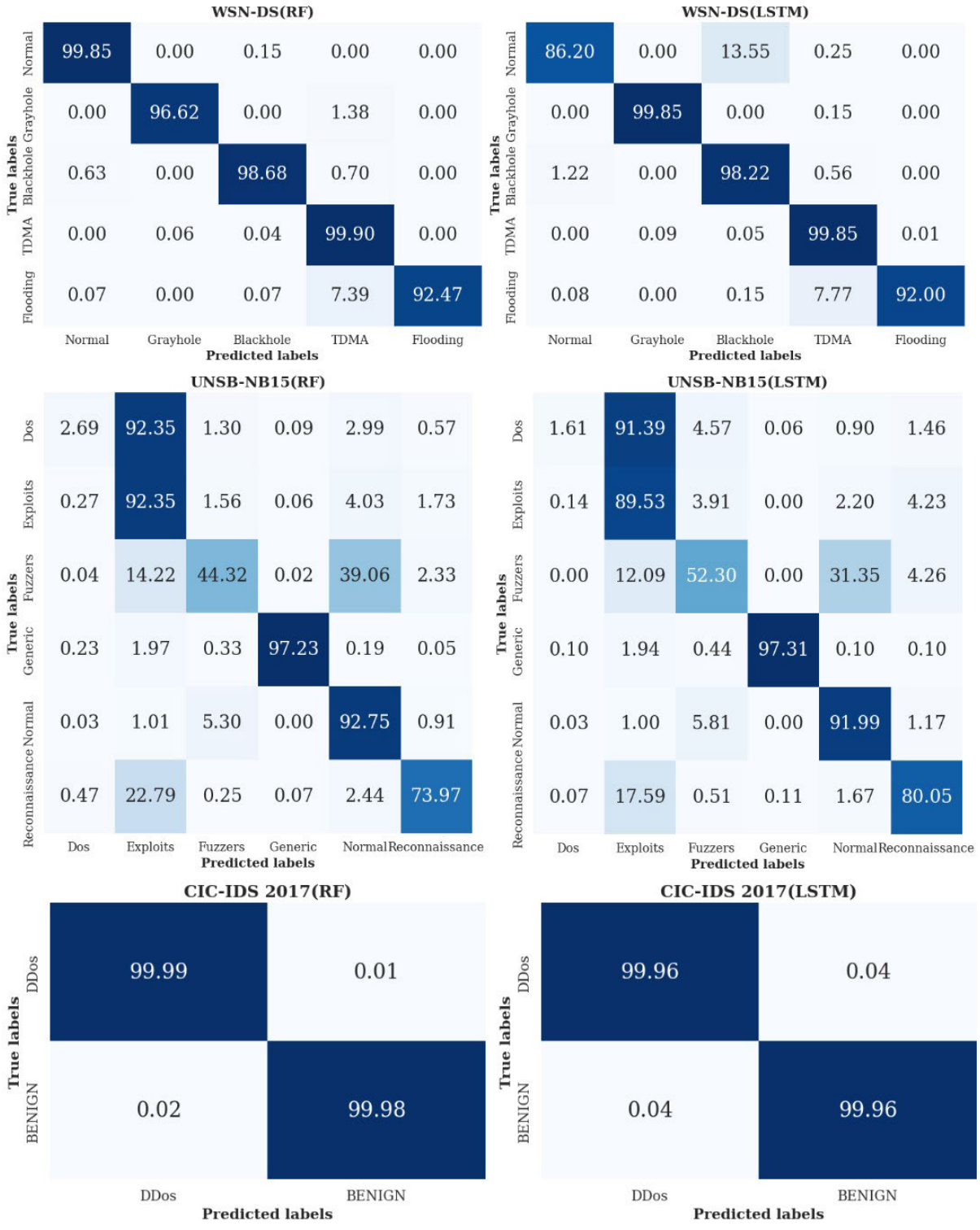


FIGURE 8. CM of RF and LSTM on WSN-DS, UNSW-NB15, and CIC-IDS 2017.

- **RF:** Training time was approximately *92 seconds*, with inference time averaging *3 milliseconds per sample*. RF's lower computational demands make it ideal for quick decision-making scenarios.
- **Long Short-Term Memory (LSTM):** Training time was significantly higher at *37 minutes*, with inference time averaging *18 milliseconds per sample*. LSTM's sequential nature allows it to capture

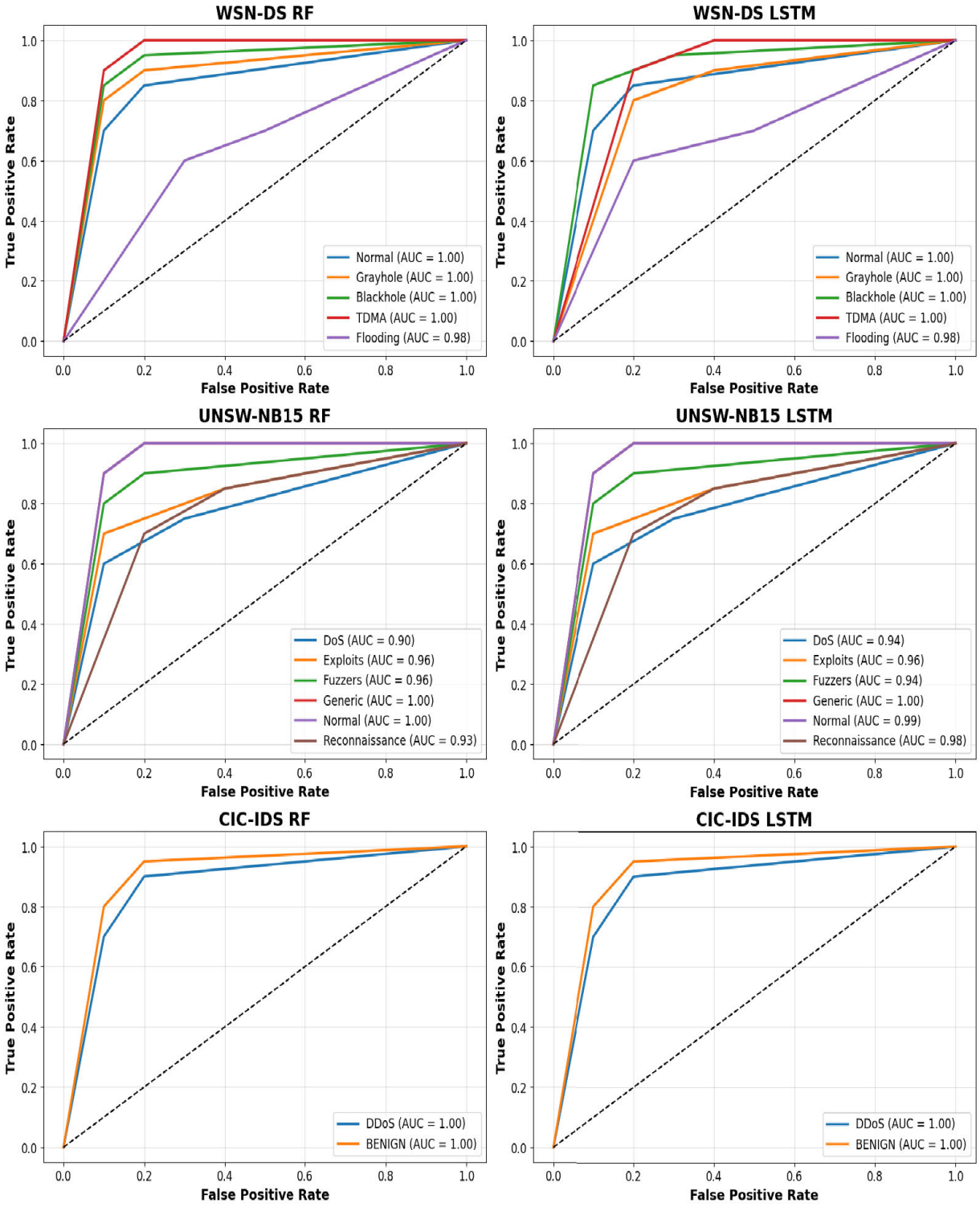


FIGURE 9. ROC of RF and LSTM on WSN-DS, UNSW-NB15, and CIC-IDS 2017.

temporal dependencies effectively, albeit with higher computational complexity.

These results highlight the trade-offs that RF excels in computational efficiency while LSTM offers enhanced capability for capturing temporal patterns in data. This discussion underscores the applicability of each model for real-time environments depending on the specific requirements of the task.

2) ASSESSMENT OF PROPOSED AND SOTA APPROACH

To address the second research question, we juxtapose the outcomes of the proposed methodology with the SOTA

TABLE 9. Comparison of proposed and SOTA approach (in %).

Model	WSN-DS				UNSW-NB15				CIC-IDS 2017			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
LSTM	99.41	99.43	99.41	99.41	82.58	82.44	82.58	80.86	99.96	99.95	99.95	99.94
RF	99.7	99.71	99.71	99.7	90.17	90.14	90.17	90.14	99.99	99.99	99.99	99.99
SOTA					84	83.2	83.4	83				
LSTM (Balanced)	98.92	98.92	98.92	98.92	80.57	83.55	80.06	80.22	99.95	99.95	99.96	99.95
RF (Balanced)	99.96	99.97	99.97	99.97	98.83	98.85	98.84	98.83	99.99	100	100	100
SOTA (Balanced)					91	91.8	91	91.2				

approach [32]. As illustrated in Table 9, the proposed method exhibits an average accuracy of 90.17% and 98.83% for default and balanced UNSW-NB15 datasets, respectively, outperforming the SOTA by 7.34% and 8.61%, correspondingly. Furthermore, it's significant to point out that while the SOTA approach focuses on a single dataset, our proposed method introduces a comparative analysis to handle multiple datasets, thereby offering a more comprehensive and adaptable solution to intrusion detection across various scenarios. It enhances feature filtering accuracy by implementing *PCC* with a higher correlation threshold of 0.97, surpassing conventional thresholds. This reduces training time complexity and boosts model accuracy by up to 3% compared to lower threshold models. Table 9 provides detailed multi-dataset classification evaluation metrics. The suggested method enhances classification performance metrics for imbalanced and balanced datasets. This analysis underscores the superiority of the proposed method over the existing SOTA approach. As mentioned in Section I, DL is more advanced and powerful for selecting features. Still, ML models can also identify crucial features for network intrusion detection based on the importance scores associated with each feature. They excel especially on small to large datasets and are lighter than DL models, which struggle with large datasets. Because RF doesn't require substantial amounts of data, they can be more effective than DL models. Through ensemble learning, RFs enhance performance by mitigating the risks associated with overfitting by uniting several weak learners to build a strong model. This model is also robust to outliers, efficient in managing noisy or irrelevant attributes, and capable of managing non-linear relationships without significant feature engineering, making it a practical option in scenarios with limited resources.

3) M/D LEARNING MODEL'S PERFORMANCE COMPARISON

In response to the second research question, a comprehensive performance comparison between M/D learning models for NID is conducted across three distinct datasets: WSN-DS, UNSW-NB15, and CIC-IDS 2017, as shown in Table 10. All models used for comparison are trained and evaluated under identical conditions. A grid search approach is employed to identify the optimal hyperparameters for all models, ensuring consistency and fairness in the comparison. Accuracy on a validation set is used as the primary metric for selecting hyperparameters, and early stopping is implemented for neural network models such as LSTM to prevent overfitting. The models are configured with carefully selected hyperparameters to ensure a fair comparison across all datasets (WSN-DS, UNSW-NB15, and CIC-IDS2017). For neural networks (LSTM, RNN, CNN), configurations included 1 to 3 layers, hidden units of 64, 128, or 256, batch sizes of 32, 64, or 128, and learning rates of 0.001, 0.0005, or 0.0001. Tree-based models (RF, DT, AdaBoost) were tuned with 50, 100, or 200 estimators and maximum tree depths of None, 10, or 20. Logistic regression used a regularization parameter (C) with values of 0.01, 0.1, 1.0, and 10. These settings are uniformly applied to ensure consistency and accuracy in the evaluation process.

The optimal configurations for LSTM best setup include 2 layers, 128 units per layer, and a sequence length of 20 timesteps. The RNN configuration uses 1 layer with 64 units and 10 timesteps. For CNN, the optimal setting involves 3 layers with 128 units. The Random Forest model is set with 50 estimators and no maximum depth. The Decision Tree also uses no maximum depth. For AdaBoost, the ideal configuration involves 100 estimators, while Bernoulli Naive Bayes performs best with 10 estimators. The K-Nearest Neighbors model operates with k=3, and the Logistic Regression model is configured with a regularization parameter C=1.0. These settings represent the best-performing configurations for each respective model.

From DL models, LSTM model exhibits outstanding accuracy, precision, recall, and F1-score on all three datasets, with notable values of 99.41%, 99.43%, 99.41%, and 99.41% for WSN-DS, 82.58%, 82.44%, 82.58%, and 80.86% for UNSW-NB15, and 99.96%, 99.95%, 99.95%, and 99.94% for CIC-IDS 2017. RNN and CNN models also showcase strong performance, with RNN achieving accuracy values of 99.12%, 82.33%, and 99.61%, and CNN achieving accuracy values of 98.66%, 82.1%, and 96.22% on WSN-DS, UNSW-NB15, and CIC-IDS 2017, respectively. In the WSN-DS dataset, the Recurrent Neural Network (RNN) achieves outstanding performance with accuracy, precision, recall, and F1-score of 99.12%, 99.22%, 99.19%, and 99.19%, respectively. Similarly, on the UNSW-NB15 dataset, RNN

TABLE 10. M/D learning model's performance comparison (in %).

Model	WSN-DS				UNSW-NB15				CIC-IDS 2017			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
LSTM	**99.41**	**99.43**	**99.41**	**99.41**	**82.58**	**82.44**	**82.58**	**80.86**	**99.96**	**99.95**	**99.95**	**99.94**
RNN	99.12	99.22	99.19	99.19	82.33	81.88	82.33	79.59	99.61	99.63	99.52	99.6
CNN	98.66	98.76	98.67	98.68	82.1	81.11	82.1	80.11	96.22	96.22	96.22	96.22
RF	**99.7**	**99.71**	**99.71**	**99.7**	**90.17**	**90.14**	**90.17**	**90.14**	**99.99**	**99.99**	**99.99**	**99.99**
DT	99.53	99.53	99.53	99.53	88.87	89.13	88.87	88.99	99.98	99.97	99.97	99.98
AdaBoost	97.34	97.99	97.34	97.53	85.97	85.4	85.97	85.6	99.98	99.99	99.99	99.99
BNB	94.23	95.74	94.23	94.42	71.28	70.42	71.28	68.96	98.33	98.37	98.33	98.33
KNN	99.54	99.55	99.54	99.54	68.67	69.84	68.67	68.79	99.96	99.96	99.96	99.96
LR	97.39	97.46	97.39	97.39	37.78	25.81	37.78	29.62	99.86	99.87	99.87	99.87

TABLE 11. Influence of filter on RF and LSTM performance (in %).

	Influence of Filter RF (In %)					Influence of Filter LSTM (In %)				
Dataset / Filter	Accuracy	Precision	Recall	F1 Score	TT	Accuracy	Precision	Recall	F1 Score	TT
WSN-DS (Enable)	99.71	99.71	99.71	99.71	508.2s	99.41	99.43	99.41	99.41	1900.85s
UNSW_NB15 (Enable)	90.17	90.14	90.17	90.14	307.3s	82.58	82.44	82.58	80.86	1000.87s
CIC-IDS 2017 (Enable)	99.99	99.99	99.99	99.99	299.9s	99.96	99.95	99.95	99.94	900.88s
WSN-DS (Disable)	99.67	99.66	99.67	99.66	522.2s	99.39	99.38	99.4	99.39	2000.25s
UNSW_NB15 (Disable)	90.15	90.1	90.14	90.11	325.2s	82.51	82.41	82.55	80.82	1020.78s
CIC-IDS 2017 (Disable)	99.96	99.96	99.99	99.96	310.2s	99.95	99.95	99.95	99.92	950.69s

TABLE 12. Influence of re-sampling on RF and LSTM performance (in %).

	Influence of Re-sampling RF (In %)				Influence of Re-sampling LSTM (In %)			
Dataset	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Undersampling								
WSN-DS	98.12	98.2	98.13	98.12	91.36	91.55	91.36	91.34
UNSW_NB15	81.66	81.71	81.66	81.68	78.18	81.15	78.19	78.48
CIC-IDS 2017	99.99	99.99	99.99	99.99	99.91	99.91	99.91	99.91
Default								
WSN-DS	99.71	99.71	99.71	99.71	99.41	99.43	99.41	99.41
UNSW_NB15	90.17	90.14	90.17	90.14	82.58	82.44	82.58	80.86
CIC-IDS 2017	99.99	99.99	99.99	99.99	99.96	99.95	99.95	99.94
Oversampling								
WSN-DS	99.96	99.97	99.97	99.97	98.92	98.92	98.92	98.92
UNSW_NB15	98.83	98.85	98.84	98.83	80.57	83.55	80.06	80.22
CIC-IDS 2017	99.99	100	100	100	99.95	99.95	99.96	99.95

TABLE 13. Model performance on clean data vs noisy data (in %).

Dataset	Accuracy (Clean)	Accuracy (Noisy)	Precision (Clean)	Precision (Noisy)
WSN-DS	99.7	99.41	99.71	99.43
UNSW_NB15	90.17	89.17	90.14	89.12
CIC-IDS 2017	99.99	99.84	99.99	99.95

demonstrates commendable metrics, achieving an accuracy of 82.33%, precision of 81.88%, recall of 82.33%, and an F1-score of 79.59%. Furthermore, on the CIC-IDS 2017 dataset, RNN excels, achieving an accuracy of 99.61%, precision of 99.63%, recall of 99.52%, and an F1-score of 99.6%. On the WSN-DS dataset, the Convolutional Neural Network (CNN) exhibits impressive performance, attaining an accuracy, precision, recall, and F1-score of 98.66%, 98.76%, 98.67%, and 98.68%, respectively. In the UNSW-NB15 dataset, CNN maintains strong metrics with an accuracy of 82.1%, precision of 81.11%, recall of 82.1%, and an F1-score of 80.11%. Furthermore, on the CIC-IDS 2017 dataset, CNN continues its robust performance, achieving an accuracy of 96.22%, precision of 96.22%, recall of 96.22%, and an F1-score of 96.22%. Machine learning methods like Decision Tree (DT), AdaBoost, Bernoulli Naive Bayes (BNB), K-Nearest

Neighbors (KNN), and Logistic Regression (LR) concerning their ability to classify results as shown in Table 10.

RF, in particular, emerges as a standout performer, showcasing exceptional performance across all three datasets (WSN-DS, UNSW-NB15, and CIC-IDS 2017). Moreover, the model can be very useful when resources are limited. It manages nonlinear connections without requiring substantial feature engineering, is robust against outliers, and quickly manages noisy or irrelevant data. The superiority of ML models can be attributed to their adaptability to diverse datasets, resilience to limited labeled data, interpretability, computational efficiency, and robustness to outliers. In contrast, DL models, represented by LSTM, RNN, and CNN, demonstrate competitive but comparatively lower performance across the evaluated metrics. Overall, the technical advantages of ML models highlight their effectiveness within the framework of network intrusion detection in comparison with to DL counterparts.

4) INFLUENCE OF FILTERING FEATURES

To address the third research query, two separate situations are analyzed to determine the effect of the filter-based method on the training duration of the proposed model. The initial situation involved implementing the filter-based technique (PCC). On the other hand, the second scenario excluded the filtering component. Table 11 provides insights into the training time (TT) durations, in seconds, for RF and LSTM models across different datasets. With or without the filter, RF displays faster training speeds than LSTM in all datasets, which have complex recurrent computations. Activating the filter significantly boosts the performance of RF and LSTM models across different datasets. The PCC filter carefully selects features closely linked to what we want to predict while eliminating noisy or irrelevant data. As a result, RF and LSTM models get a clearer picture of the data, making them better at distinguishing patterns and making predictions. This improvement shows up in accuracy, precision, recall, and F1 score metrics. Also, the filter helps prevent overfitting, where models memorize noise instead of learning real patterns. By focusing on the most important features, the filter makes the models better at handling new, unseen data. Additionally, it makes training the models faster because it reduces the amount of data they need to process. Overall, using the filter with RF and LSTM models makes them better at predicting outcomes by focusing on important features, eliminating noise, and working more efficiently.

5) INFLUENCE OF RE-SAMPLING

Resampling can change a dataset's class distribution and adjust bias. The fourth research question examines the impact of resampling using oversampling and undersampling. The insignificant class is oversampled with random oversampling, whereas the significant class is undersampled with random selection. Table 12 presents the impact of re-sampling techniques on the performance of RF and LSTM models across different datasets. RF achieved accuracy, precision, recall, and F1 score values of 98.12%, 98.2%, 98.13%, and 98.12%, respectively, on undersampled WSN-DS. LSTM, under the same conditions, recorded 91.36%, 91.55%, 91.36%, and 91.34%. RF achieved 99.71% accuracy on WSN-DS, whereas LSTM achieved 99.41% accuracy on default WSN-DS. RF achieved 99.96% accuracy, while LSTM achieved 98.92% accuracy on oversampled WSN-DS. Similar trends are observed across UNSW_NB15 and CIC-IDS 2017 datasets. Undersampling generally results in lower performance, default settings show stable performance, and oversampling enhances RF's performance. LSTM, on the other hand, shows varying performance trends across different datasets and re-sampling techniques.

LSTM's performance varies with re-sampling due to dataset-specific factors like complexity and class imbalances. Adaptability is influenced by how re-sampling alters data distribution and sequential patterns, necessitating tailored approaches for each dataset.

6) ROBUSTNESS EVALUATION TO NOISY DATA AND ABNORMAL SAMPLES

To assess the robustness of the proposed model to noisy data and abnormal samples, this study conducted experiments by introducing three types of noise into the dataset, namely Gaussian noise, outliers, and label noise using the RF model. The model's performance is evaluated on these noisy datasets and compared to the performance on the clean data. The results show that the model remains robust to Gaussian noise, with only a minimal drop in accuracy (up to 0.3%) and precision. However, the model exhibited more noticeable degradation when outliers were introduced, with up to a 1% drop in performance, suggesting sensitivity to extreme values. Label noise caused the most significant performance degradation, dropping accuracy by as much as 1.5%. Despite these challenges, the model's performance on noisy data remains impressive, with only a slight reduction in key metrics. These findings indicate that the model effectively handles real-world noise but could benefit from further refinements to improve robustness against extreme outliers and label noise.

V. CONCLUSION AND FUTURE WORK

This study aims to develop an effective network IDS by integrating dataset preprocessing and feature extraction using PCC. A range of M/DL models, including LSTM, RNN, CNN, RF, DT, Adaboost, BNB, KNN, and LR, are employed to identify the most effective intrusion detection approach. Performance metrics are used to evaluate the efficiency of attack detection algorithms across three distinct datasets: WSN-DS, UNSW-NB15, and CIC-IDS 2017. The findings reveal that the RF model from ML and the LSTM model from DL achieved the highest accuracy rates with the selected features. Furthermore, the RF model showed greater consistency and strength in accuracy and training duration than the LSTM model.

Enhancing network intrusion detection systems involves several key strategies. These include improving feature engineering, utilizing ensemble learning for better performance, and creating adaptable models to address evolving threats. Prioritizing clarity in deep learning models, adopting real-time strategies, and exploring transfer learning is also crucial. Extended evaluations, cloud deployment, and collaborative defences aim to further strengthen network security against emerging threats.

REFERENCES

[1] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Secur. Commun. Netw.*, vol. 2020, pp. 1–11, Aug. 2020.

[2] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022.

[3] H. Alkahtani and T. H. H. Aldhyani, "Intrusion detection system to advance Internet of Things infrastructure-based deep learning algorithms," *Complexity*, vol. 2021, no. 1, pp. 1–18, Jan. 2021.

[4] M. Ozkan-Okay, R. Samet, Ö. Aslan, and D. Gupta, "A comprehensive systematic literature review on intrusion detection systems," *IEEE Access*, vol. 9, pp. 157727–157760, 2021.

[5] A. Patel, H. Alhussian, J. M. Pedersen, B. Bounabat, J. C. Júnior, and S. Katsikas, "A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems," *Comput. Secur.*, vol. 64, pp. 92–109, Jan. 2017, doi: 10.1016/j.cose.2016.07.002.

[6] B. Sharma, L. Sharma, and C. Lal, "Anomaly detection techniques using deep learning in IoT: A survey," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Dec. 2019, pp. 146–149, doi: 10.1109/ICCIKE47802.2019.9004362.

[7] P. Wu, "Deep learning for network intrusion detection: Attack recognition with computational intelligence," Ph.D. dissertation, UNSW Sydney, Sydney, NSW, Australia, 2020.

[8] D. E. Denning, "Intrusion detection revisited," *ACM Trans. Inf. Syst. Secur. (TISSEC)*, vol. 5, no. 4, pp. 412–417, 1987.

[9] M. Putchala, "Deep learning approach for intrusion detection system 687 (IDS) in the Internet of Things (IoT) network using gated recurrent neural 688 networks (GRU)," Ph.D. dissertation, Dept. Comput. Sci. Eng., Wright State Univ., Dayton, OH, USA, 2017.

[10] R. K. Vigneswaran, R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in *Proc. 9th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2018, pp. 1–6.

[11] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model," *Complexity*, vol. 2021, no. 1, pp. 1–11, Jan. 2021.

[12] H. Mankodiya, M. S. Obaidat, R. Gupta, and S. Tanwar, "XAI-AV: Explainable artificial intelligence for trust management in autonomous vehicles," in *Proc. Int. Conf. Commun., Comput., Cybersecurity, Informat. (CCCI)*, Oct. 2021, pp. 1–5.

[13] P. Lin, K. Ye, and C.-Z. Xu, "Dynamic network anomaly detection system by using deep learning techniques," in *Proc. 12th Int. Conf., Held as Part Services Conf. Fed.*, San Diego, CA, USA. Cham, Switzerland: Springer, Jun. 2019, pp. 161–176.

[14] A. Meliboev, J. Alikhanov, and W. Kim, "Performance evaluation of deep learning based network intrusion detection system across multiple balanced and imbalanced datasets," *Electronics*, vol. 11, no. 4, p. 515, Feb. 2022.

[15] T. H. Hai and L. H. Nam, "A practical comparison of deep learning methods for network intrusion detection," in *Proc. Int. Conf. Electr., Commun., Comput. Eng. (ICECCE)*, Jun. 2021, pp. 1–6.

[16] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system," in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2017, pp. 167–172.

[17] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," *J. Netw. Comput. Appl.*, vol. 177, Mar. 2021, Art. no. 102942.

[18] R. R. Reddy, K. A. Reddy, C. M. Kumar, and Y. Ramadevi, "Detection of network anomaly sequences using deep recurrent neural networks," in *Proc. 4th Int. Conf. Smart Comput. Informat.*, vol. 2. Cham, Switzerland: Springer, 2021, pp. 605–615.

[19] S. Hosseininoorbin, S. Layeghy, M. Sarhan, R. Jurdak, and M. Portmann, "Exploring edge TPU for network intrusion detection in IoT," *J. Parallel Distrib. Comput.*, vol. 179, Sep. 2023, Art. no. 104712.

[20] M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, no. 5, p. 834, May 2021.

[21] I. Debicha, R. Bauwens, T. Debatty, J.-M. Dricot, T. Kenaza, and W. Mees, "TAD: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 138, pp. 185–197, Jan. 2023.

[22] B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "Network intrusion detection model based on CNN and GRU," *Appl. Sci.*, vol. 12, no. 9, p. 4184, Apr. 2022.

[23] Z. Wang, K. W. Fok, and V. L. L. Thing, "Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study," *Comput. Secur.*, vol. 113, Feb. 2022, Art. no. 102542.

[24] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, and R. Zhang, "Model of the intrusion detection system based on the integration of spatial–temporal features," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101681.

[25] S. B. Bastola, S. Shakya, and S. Sharma, "Distributed denial of service attack detection on software defined networking using deep learning," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Udupi, India, 2021, pp. 13–16.

[26] P. Zhao, Z. Fan, Z. Cao, and X. Li, "Intrusion detection model using temporal convolutional network blend into attention mechanism," *Int. J. Inf. Secur. Privacy*, vol. 16, no. 1, pp. 1–20, Oct. 2021.

[27] M. Aldwairi and D. Alansari, "N-grams exclusion and inclusion filter for intrusion detection in Internet of Energy big data systems," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, p. 3711, Mar. 2022.

[28] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, p. 898, Mar. 2022.

[29] P. Rajesh Kanna and P. Santhi, "Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features," *Knowl.-Based Syst.*, vol. 226, Aug. 2021, Art. no. 107132.

[30] P. R. Kanna and P. Santhi, "Hybrid intrusion detection using MapReduce based black widow optimized convolutional long short-term memory neural networks," *Exp. Syst. Appl.*, vol. 194, May 2022, Art. no. 116545.

[31] W. Wang, S. Jian, Y. Tan, Q. Wu, and C. Huang, "Robust unsupervised network intrusion detection with self-supervised masked context reconstruction," *Comput. Secur.*, vol. 128, Feb. 2023, Art. no. 103131.

[32] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," *Comput. Electr. Eng.*, vol. 107, Apr. 2023, Art. no. 108626.

[33] W. D. Xiong, K. L. Luo, and R. Li, "AIDTF: Adversarial training framework for network intrusion detection," *Comput. Secur.*, vol. 128, May 2023, Art. no. 103141.

[34] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, pp. 1–16, Jan. 2016.

[35] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine learning and deep learning approaches for CyberSecurity: A review," *IEEE Access*, vol. 10, pp. 19572–19585, 2022.

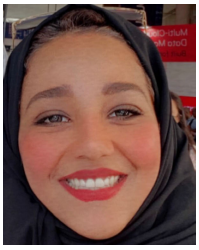
[36] B. Sharma, L. Sharma, and C. Lal, "Feature selection and deep learning technique for intrusion detection system in IoT," in *Proc. Int. Conf. Comput. Intell.* Cham, Switzerland: Springer, 2022, pp. 253–261.

[37] A. K. Al Hwaitat, M. A. Almaiah, O. Almomani, M. Al-Zahrani, R. M. Al-Sayed, R. M. Asaifi, K. K. Adhim, A. Althunibat, and A. Alsaaidah, "Improved security particle swarm optimization (PSO) algorithm to detect radio jamming attacks in mobile networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, pp. 176495–176520, 2020.

[38] M. N. Khan, H. U. Rahman, M. A. Almaiah, M. Z. Khan, A. Khan, M. Raza, M. Al-Zahrani, O. Almomani, and R. Khan, "Improving energy efficiency with content-based adaptive and dynamic scheduling in wireless sensor networks," *IEEE Access*, vol. 8, pp. 176495–176520, 2020.

[39] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random forest," in *Proc. 3rd Int. Conf.*, Chengde, China. Cham, Switzerland: Springer, Sep. 2012, pp. 246–252.

[40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[42] T. Saito and M. Rehmsmeier. (2017). *Basic Evaluation Measures From the Confusion Matrix*. [Online]. Available: https://classeval.wordpress.com/introduction/basic-evaluation-measures



HANADI HAKAMI is a highly skilled Software Development Researcher. She works well under pressure and consistently meets deadlines and targets while delivering high-quality work. She has an excellent problem-solving approach and a proven ability to generate fresh solutions. She has developed and published various new models in high-impact journals and IEEE peer-reviewed conferences.

MUHAMMAD FAHEEM is a Senior Scientist at VTT Technical Research Centre of Finland. He has held various academic positions at Abdullah Gül University, Turkey, COMSATS University, Pakistan, and Universiti Teknologi Malaysia, from 2012 to 2022. From 2022 to 2024, he served as an Assistant Professor, Program Manager for the Master's in Robotics, and Academy of Finland Researcher at the University of Vaasa. He has received several prestigious awards, including the Top 2% Scientist Award from Stanford University (2023–2024), the Highly Cited Scientist Award from Elsevier (2024), and the Top 1% Turkish Scientist in the World Award (2021). He is actively involved in editorial boards of reputed journals such as *Sustainable Futures* (Elsevier), PLOS ONE, *Frontiers in the Internet of Things*, *Frontiers in Artificial Intelligence*, and *Computers, Materials and Continua*, among others. He has also served as a Lead Guest Editor for special issues on AI, ML, and BC technologies in leading journals, including IET, Wiley, and MDPI. He is a reviewer for prestigious publications by ACM, IEEE, Springer, Wiley, and Elsevier. He has chaired sessions at several IEEE conferences, including AIE2024 (Finland) and CAC2023 (Malaysia). Additionally, he is an active member of IEEE, ERIGrid 2.0, the JUFO panel (evaluating journal quality and standards), and AI-Doc (evaluating doctoral students) in Finland. His research expertise spans cybersecurity, AI, ML, BC, SGs, and IoT. He specializes in designing, modeling, developing, and piloting ML and BC-based solutions for smart grid applications.

MAJID BASHIR AHMAD received the master's degree in computer science from COMSATS University Islamabad, Pakistan, in 2014, and the M.S. degree in computer science from The University of Lahore, Pakistan, in 2019. He is currently a Research Scholar in the field of computer science. His research interests include artificial intelligence, machine learning, and data mining.