Karuna Adhikari

Comparative Analysis for Electricity Load Forecasting using Machine Learning Models

School of Technology and Innovation
Master's thesis in Sustainable and Autonomous System

Vaasa 2026

UNIVERSITY OF VAASA
School of Technology and Innovation
Author: Karuna Adhikari
Title of the Thesis: Comparative Analysis for Electricity Load Forecasting using Machine Learning Models
Degree: Master of Science in Technology
Programme: Sustainable and Autonomous System
Supervisor: Mohammed Elmusrati
Instructor: Petri Välisuo
Year: 2026 Pages: 67

ABSTRACT:
Electricity load forecasting has become more complex in the modern power system as varieties of renewable energy generation; plug-in electric vehicles and demand response are increasing in the energy mix. Reliable demand forecasting is critical for generation planning, reserve provision, electricity system stability, optimal operation and trading in electricity markets. Finland with strong seasonal cycles due to the country's climate and ties with Nordic electricity market makes load forecasting crucial for operational and economic efficiency.

While there is a vast pool of literature available on application of machine learning for electricity load forecasting, direct comparison across models is unreliable because of varying experimental setup, preprocessing techniques and typical evaluation at a single time horizon. This thesis aims to address the shortcomings by comparing six different machine learning models in a controlled way by using Finnish electricity load values and drawing the impact of forecasting horizon on the model's performance. This study draws on the published literature on machine learning, deep learning, ensemble learning, recurrent neural network systems and convolutional feature extraction techniques.

The six models were tested on the data obtained from two open Finnish datasets. First one being hourly national electricity demand statistics released by Fingrid and another one being hourly meteorological data from Finnish Meteorological Institute (FMI) from the year 2025 and 2026. All the models utilize the same feature engineering process known as lagged demand terms, rolling statistical measures, cyclical calendar information, trigonometric terms of the Fourier series and meteorological variables. RMSE, MAE, MAPE and SMAPE assess model performance across short-term, medium-term and long-term absolute time horizons.

The finding suggests that the Random Forest model has delivered strongest overall predictive performance across all three forecasting horizons and all error metrics. It significantly exceeds the accuracy of both classical statistical models and other deep learning architectures. The more competitive deep learning model has been the Convolutional Neural Network while the Long Short-Term Memory model has exhibited worst-case performance failing to even outperform the linear model trendline. Relative model rankings have been observed to be consistent across each of the three forecasting horizons, reflecting the nature of the load signal at each timescale.

KEYWORDS: electricity load forecasting, machine learning, Random Forest, LSTM, Convolutional Neural Network, deep learning, forecasting horizon, feature engineering, FMI.

Contents

1 Introduction 9
1.1 Background and Motivation 9
1.2 Research Context 10
1.3 Problem Statement 11
1.4 Aim and Research Questions 12
1.5 Scope and Limitations 13
1.6 Thesis Structure 14
2 Literature Review 15
2.1 Electricity Load Forecasting Overview 15
2.2 Traditional Statistical Methods 16
2.2.1 ARIMA and Box-Jenkins Framework 16
2.2.2 Structural Models and Adaptive Filtering 17
2.2.3 Exponential Smoothing and Semi-Parametric Models 18
2.3 Machine Learning for Load Forecasting 19
2.3.1 Artificial Neural Networks (ANN) 19
2.3.2 Support Vector Machines (SVM) 20
2.3.3 Random Forest 21
2.4 Deep Learning Architectures 22
2.4.1 Long Short-Term Memory Networks 22
2.4.2 Convolutional Neural Networks 24
2.4.3 Hybrid CNN-LSTM Architecture 25
2.5 Feature Engineering and Data Quality 26
2.6 Forecasting Horizons and Model Suitability 27
2.7 Comparative Summary Table 29
2.8 Thesis Objectives and Research Questions 30
3 Methodology 32
3.1 Datasets 32
3.2 Data Preprocessing 32
3.2.1 Timestamp Alignment and Unit Conversion 33

3.2.2 Handling Missing Values 33
3.2.3 Dataset Merging 33
3.2.4 Outlier Detection and Clipping 33
3.2.5 Feature Scaling 34
3.3 Feature Engineering 35
3.4 Model Architecture and Configuration 36
3.4.1 Linear Regression 37
3.4.2 Support Vector Regression 37
3.4.3 Random Forest 37
3.4.4 LSTM Network 38
3.4.5 Convolutional Neural Network 38
3.4.6 Hybrid CNN-LSTM Architecture 38
3.5 Hyperparameter Tuning and Training Protocols 39
3.6 Evaluation Metrics 39
3.7 Experimental Setup 40
3.7.1 Chronological Train-Validation-Test Split 41
3.7.2 Controlled Comparison Conditions 41
3.7.3 Computational Resource Documentation 41
3.8 Forecasting Horizons 42
4 Result and Analysis 43
4.1 Data Preprocessing and Feature Analysis 43
4.2 Overall Model Performance 44
4.3 Horizon-stratified Evaluation 45
4.3.1 Short-term Horizon 46
4.3.2 Medium-term Horizon 47
4.3.3 Long-term Horizon 47
4.4 Training Deep Learning Models 47
4.4.1 CNN Training and Architecture 47
4.4.2 CNN-LSTM Architecture and Training 49
4.4.3 LSTM and Convergence 50

4.5 Summary of Results 51

5 Discussion 53

5.1 Research Question 1: Which is the Most Accurate Electricity Load Forecasting Model for Finland? 53

5.2 Research Question 2: Forecast Horizon: Unfolding of Model Performance 54

5.3 Research Question 3: Performance vs Efficiency 55

5.4 Comparison with Literature Review 56

5.5 Methodological Limitations 58

5.5.1 Size and Time Period of the Dataset 58

5.5.2 Single Weather Station 58

5.5.3 No Hyperparameter Optimization of Deep Learning 58

5.5.4 Point Forecasting Only 59

5.5.5 SVR Computational Restriction 59

5.6 Future Work 59

6 Conclusion 61

References 64

Appendix 1- Full Source Code 67

Figures

Figure 1. LSTM Cell Architecture 23

Figure 2. CNN-LSTM Structure 25

Figure 3. Model Suitability by Forecasting Horizon 29

Figure 4. Methodology Pipeline 31

Figure 6. Bar graph for all model comparisons 45

Figure 7. Horizon-stratified evaluation across short-term, medium-term and long-term windows 46

Figure 8. CNN training history 48

Figure 9. CNN first 200 test hours 49

Figure 10. CNN-LSTM first 200 test hours 50

Figure 11. LSTM training history 50

Tables

Table 1. Comparative summary of electricity load forecasting 29

Table 2. Dataset Summary 32

Table 3. Summary of feature engineering 35

Table 4. Model Architecture and Hyperparameters 36

Table 5. Evaluation Metrics 40

Table 6. Forecasting Horizon and Evaluation Windows 42

Table 7. Correlations between load and weather variables 43

Table 8. Model Comparisons 44

Table 9. Horizon-based RMSE summary and comparison with literature expectations 55

Abbreviations

ML	Machine Learning
LR	Linear Regression
RF	Random Forest
SVR	Support Vector Regression
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
ARIMA	Autoregressive Integrated Moving Average
ANN	Artificial Neural Network
FMI	Finnish Meteorological Institute
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
SMAPE	Symmetric Mean Absolute Percentage Error
IQR	Interquartile Range
OLS	Ordinary Least Square
MSE	Mean Square Error
FEA	Finnish Energy Authority

1 Introduction

1.1 Background and Motivation

Forecasting electricity demand is among the most important operations of power systems. Load forecasts are critical for power system operators, energy marketing agents and infrastructure planners to provide adequate generation scheduling, fuel procurement, reserve margins and solve the real-time supply-demand balance problem. Accurate forecasts result in efficient use of generation assets, minimal congestion on transmission network and fair pricing of energy in wholesale markets. Inaccurate forecasts can lead to expensive reserves, power system dynamics and in some cases involuntary load shedding.

Over the last decade, the task of making accurate forecasts has become more difficult. The use of various renewable energy sources such as wind and solar photovoltaic to supply electricity introduces a source of uncertainty which was not present in the largely dispatchable generation mixes of decade past. The nature of demand has also evolved in parallel to rapid rise of electric vehicles bringing in large pulse charging loads that are concentrated in time. Overall, these changes bring a demand shield that is non-linear, non-stationary and harder to capture with the traditional statistical techniques that prevailed in 90s.

Finland provides an important load forecasting test bed. The climate plays a significant role in the Finnish electricity demand which is correlated to the space heating demand and the national load peaks above 14,000 MWh per hour in the winter that drops below 6,000 MWh per hour in the spring and autumn. This strong temperature dependent seasonality along with the weekly and daily periodicities reflecting industrial and household activity patterns give rise to a complex but structured load forecasting challenge.

The use of machine learning for electricity load forecasting has gained high attention since the pioneering work of Park et al. (1991) who showed that feedforward neural

networks were able to adapt to non-linear load-weather relationships more successfully than linear statistical methods. Years of follow up research have seen success in Support Vector Regression, ensemble methods like Random Forest and more recently on deep learning models such as Long Short-Term Memory, Convolutional Neural Networks and even hybrid variants. The overall message here is that more expressive models can outperform traditional statistical methods like ARIMA and exponential smoothing to produce better forecast when carefully applied.

But a practically relevant issue often undermines the trust of many studies. The different studies compare different models on different data with different preprocessing procedures, feature selection and forecast evaluation criteria. A model that is deemed best in one study might not be best in another due to difference in experimental setup. This makes it hard to address a practical question of "if I have a particular dataset, a particular set of features and a particular forecasting horizon which model should I use?". The current thesis seeks to resolve this situation by training and comparing six model architectures on the same data using the same processing pipeline and comparing performance for three operationally relevant forecasting horizons.

1.2 Research Context

This thesis is in between among two ongoing research. The first one is the continuous creation of machine learning algorithms to predict time series which has been quickened by the availability of deep learning models and enough open data on energy. The second is the operational difficulty to Nordic grid operators, specifically Fingrid, of sustaining forecast accuracy in a power system that is shifting to a more renewable-dominated generation mix in favor of a dispatchable hydro-thermal generation mixture.

A well-documented and publicly available data environment can be found in the Finnish electricity system to carry out this study. Fingrid publicly sources hourly national load data through its open data portal and the Finnish Meteorological Institute publishes hourly surface weather observations on stations that are spaced throughout the country.

A joint result of these two sources is the possibility to build a feature rich dataset that not only characterizes the temporal autocorrelation structure of demand but also the requirement of the demand to a range of external meteorological drivers without needing to obtain proprietary operational data.

The six architecture models chosen to be compared are key families of machine learning techniques that have been used in the literature of load forecasting. Linear Regression is interpretable baseline that measures the amount of variance solely due to the engineered feature set itself and without any non-linear modelling abilities. The classical kernel machine implementation of SVM using a radial basis function kernel is said to conserve convex optimization properties and minimize structural risk. The representation of ensemble method is known as Random Forest where bootstrap aggregation and random feature sampling is done to ensure that predictive variance is limited without increasing bias. The three deep learning models named stacked LSTM, hierarchical CNN and hybrid CNN-LSTM are used for situation in sequence modeling of load forecasting.

The study adjusts all six models to use the same preprocessing, feature engineering, train-validation-test split and evaluation metrics to remove the contribution of model architecture and data preparation resulting in directly informative comparisons to researchers choosing forecasting tools in operational settings.

1.3 Problem Statement

While the number of published works about the use of machine learning in electricity load forecasting has advanced, there are three main problems that limit the general usability of this type of research in grid operations and energy system engineering.

First, there is very little confidence in comparing studies due to uncontrolled experimental setup. For instance, if two different experiments show different accuracy results it would not be clear whether the relative difference is due to an improvement in architecture or just differences in data sets, features or evaluation process. In practical sense

it leads to a lack of consistency in model selection and prevents the gathering of insights that can be replicated in different forecasting settings to gain knowledge about which architectures are consistent across a variety of forecasting horizons.

Second, most of the comparative studies consider models on one forecasting horizon. It may either be short-term or medium-term, but they never investigate how the relative model ranking varies along with the forecasting horizon. The functional elements of short-term forecasting like real-time dispatch and intraday market precipitation are too fundamentally different to those of medium-term forecasting. An optimally performing model at 24-hour horizon would be nowhere near an optimum in a study that only measures a single horizon and offers partial information to institutes that need to generate forecasts at more than one planning timescale at a time.

Third, feature engineering is not always fixed between the models in observations. Other studies use deep models on small sets of features on the assumption that these models can directly learn the representations of raw sequences whereas richer engineered features are used on classical models. The design option makes is hard to compare between the performance differences that may be observed due to the quality of the input representations instead of the architectural properties. A more controlled study that uses a shared set of features with every model gives a better test of the hypothesis that architecture is not feature quality and that it dictates the predictive accuracy of models.

1.4 Aim and Research Questions

Overall goal of this thesis is to undertake a methodologically controlled comparative analysis of six different machine learning architectures to predict the electricity load of the Finnish national grid. This is done to clearly account for the influence of the predictive horizon and the potential separation of architecture and feature engineering contributions for accuracy predictive skill.

This is categorized into three different research questions:

- RQ1: What are the most precise machine learning models to predict the electricity load in Finland with a pre-determined time horizon and standardized collection of engineered characteristics?
- RQ2: What is the relationship between the forecasting horizon and the relative as well as absolute performance of the six chosen model architecture?
- RQ3: What is the extent of a trade-off between model predictive performance and computational efficiency in the six possible model architectures?

1.5 Scope and Limitations

There are various ways that make this study limited. The dataset obtained from Fingrid and Finnish Meteorological Institute will have enough seasonal coverage to train and evaluate the model, but it will be less generalizable to other systems. The only weather data used is from Helsinki-Malmi station that is likely to be limited to represent the meteorological diversity of Finland.

Each of the six models generates deterministic point forecast and the performance is measured using four point metric scores such as Root Mean Squared Error, Mean Absolute Error, Mean Absolute Percentage Error and Symmetric Mean Absolute Percentage Error. Other probabilistic forecasting such as quantile regression, conformal prediction and Monte Carlo dropout are also not the focus of this research.

The hyperparameter choices are determined by literature rather than by grid search or Bayesian optimization. Transformer-based architectures, gradient boosting algorithms like XGBoost or LightGBM and hybrid statistical machine learning models like ARIMA-ANN mixes are not a part of this study.

1.6 Thesis Structure

The thesis is structured in the following way:
Chapter 2 offers a general literature review of the research on electricity load forecasting that includes classical statistical modeling and machine learning approaches, neural networks, feature engineering and how the forecasting horizon and model suitability correlate. The chapter is concluded with research questions that this thesis works to answer. Chapter 3 presents methodology that focuses on the source of data, preprocessing pipeline, feature engineering taxonomy, model architecture and hyperparameter specifics, evaluation metrics and experimental setup. The results are shown in Chapter 4 in the form of overall model performance, horizon-stratified evaluation and a deep learning training dynamics analysis. Chapter 5 talks about the finding within the framework of research questions and previous literature, outlines the methodological shortcomings of the study and suggests future research directions. Chapter 6 is the final part of this thesis, and it summarizes the main result and their implications on the practical level.

2 Literature Review

2.1 Electricity Load Forecasting Overview

Electricity load forecasting is an important operational activity of the current power system. The accurate determination of demand makes up the basis of generation scheduling, cost reduction and grid stability. These functionalities have become increasingly challenging along with the incorporation of renewable energy generation sources such as wind, solar, hydropower, etc., electric vehicles and smart grid infrastructures. The influence of industrial production and residential comfort determines demand, distributed photovoltaic production, car charging cycles and adaptive tariffs. This implies non-linear multiscale interactivity in demand is quite unaddressed in classical statistical methods.

There has been a long-standing investigation of machine learning (ML) designs by academic scholars which extends the initial research of the “1900s” on experimental work by using artificial neural networks (ANN) up to modern hybrid deep learning designs that merges convolutional and recurrent layers. However, the amount of published literature does not withstand, and cross model comparisons are often internalized on different sets of data and sub-data. They lack consistency in their evaluation measures or have limited scope of forecasting horizons. Therefore, where there is need of investigation into the type of model to use, there will likely be different conflicting results. The literature is extensive but fragmented: comparative studies routinely use different datasets, evaluation metrics, and preprocessing pipelines, making direct cross-model conclusions unreliable (Hippert et al., 2001; Jain & Gupta, 2024).

This literature review is divided into five thematic sections. They include: (1) traditional statistical methods, (ii) classical machine learning models, (iii)deep learning models, (iv)feature engineering and data quality, and (iv)forecasting horizons.

2.2 Traditional Statistical Methods

2.2.1 ARIMA and Box-Jenkins Framework

Electricity load forecasting statistical tradition was originally developed in early “1970s” through the research of Box and Jenkins three stage process. The traditional three stage process of ARIMA forecasting consisted of model identification, parameter estimation, and diagnostic checking. It became widely used forecasting practice over the decades. The framework catered to non-stationary by formalizing differencing methods which is an inherent property of electricity demand in case of long-term growth. It also provides strong criteria for model selection. The Box-Jenkins methodology was reliable in situations where load patterns have a high degree of autocorrelation as well as stable seasonality.

$$(1 - \varphi_1 B - \ldots - \varphi_p B^p)(1 - B)^d Y_t = (1 + \theta_1 B + \ldots + \theta q B q)\, \varepsilon_t \qquad (1)$$

where B is the backshift operator, ϕ_i are autoregressive coefficients, θ_j are moving-average coefficients, d is the order of differencing, and $\varepsilon_t \sim N(0, \sigma^2)$ is white noise.

The Box-Jenkins method offers a well-established approach for non-stationary series. It also gives formal approach to model selection either it be Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) as well as a residual analysis for model validation. This method works well if load is highly autocorrelated and shows seasonality. However, the assumption of linearity is a fundamental flaw in the present grids where demand is driven by temperature, occupancy, prices and renewable intermittency all in a non-linear way. Also, model parameters are estimated from a data sample and are fixed. Thus, ARIMA models are unable to adapt to changes in the statistical properties of the load series over time. Such limitations pushed towards the search for more accommodating methods.

2.2.2 Structural Models and Adaptive Filtering

Harvey (1991) built upon the statistical toolkit using structural time-series models which break down a demand signal into interpretable components:

$$Y_t = \mu_t + \gamma_t + \varepsilon_t \qquad (2)$$

where μ_t is a stochastic trend, γ_t is a stochastic seasonal component, and ε_t is an irregular disturbance. Those components consist of trend, seasonality and irregular fluctuations. The framework by Harvey that incorporated the Kalman filter as recursive estimator makes it possible to update real time forecasts with new observations, a feature of basic operational importance to grid dispatch.

Although these can be done, in structural models, the state equation assumes linearity limited by its use in the context of non-linear load weather relationship. Almeshaiei & Soltan (2011) then proposed a pragmatic approach in model choice by forecasting models depending on the data properties, forecasting span, and operational specifications with attention given to neglected preprocessing steps that includes outlier identification, handling missing data points, and normalization of features.

The time varying parameters are more flexible than ARIMA models. The decomposition reflects the natural components of the physical drivers of electricity demand like trend, daily and weekly cyclicity. On-line state updating allows on-line predictions without the need for re-estimation. However, the state equations remain linear which limits their use in situations where there is presence of highly non-linear load-weather relationship. Almeshaiei & Soltan (2011) concluded that there is no best statistical model and the efficiency of models is highly dependent on the problem. This idea has been reinforced in many machine learning studies.

2.2.3 Exponential Smoothing and Semi-Parametric Models

Taylor (2003) compared Holt-Winters exponential smoothing to load forecasting over a short time and proved that even basic models have the potential to perform as good as more complex models when there is seasonality in th underlying data.

$$S_t = \alpha Y_t + (1 - \alpha) S_{t-1}, \quad 0 < \alpha < 1 \quad (3)$$

where S_t is the smoothed value at time t, Y_t is the observed value at time t, S_{t-1} is the previous smoothed value, and $\alpha \in (0, 1)$ is the smoothing parameter controlling the weight given to recent observations.

With this equation he made his goal to match the complexity of a model to the complexity of the data, instead of maximizing in the hopes of higher accuracy. He also introduced intraday seasonality which is especially important with load in electricity due to highly reinforced within day periodicities.
Fan & Hyndman (2012) brought the classical statistics and machine learning closer by introducing a semi-parametric additive model:

$$Load = \beta_0 + f_1(Temperature) + f_2(Time - of - Day) + \beta_3 \cdot Calendar + \varepsilon \quad (4)$$

where β_0 is the intercept term, $f_1(\cdot)$ and $f_2(\cdot)$ are non-parametric smoothing functions capturing the non-linear effects of temperature and time-of-day respectively, β_3 is a linear coefficient for the calendar variable, and ε is the residual error term.

It consists of linear regression and non-parametric smoothing functions that allows capturing the non-linear relationships between temperature and load and still retain its interpretability. They also mostly focused on probabilistic forecasting that produces prediction intervals instead of point estimation to suggest a more general methodological change to quantification of uncertainty. The semi-parametric and

exponential smoothing gets us half way there. They offer better predictions than ARIMA when seasons are not simple while requiring significantly less computational resources. They are limited when the non-linear relationships are not sufficiently smooth.

2.3 Machine Learning for Load Forecasting

2.3.1 Artificial Neural Networks (ANN)

The history of artificial neural network's application in electricity load prediction can be tracked to begin with the work of Park et al. (1991) who demonstrated that even simple feedforward networks were able to learn non-linear interactions between load and weather variables sucessfully. A single hidden layer, least squares estimate network:

$$\hat{y} = g(W_2 \cdot \sigma(W_1 x + b_1) + b_2) \quad (5)$$

where x is the input feature vector, W_1, W_2 are weight matrices, b_1, b_2 are biases, σ is a non-linear activation (ReLU), and g is the output function.

In the early 1990s when even the computational power was extremely limited, his study proved that data-driven non-linear model could be an effective direction of the power systems community. Besides him, Zhang et al. (1998) also presented a set of comparisons which revealed that ANNs performed better than ARIMA in the non-linear context whereas it was important to note that overfitting can also be achieved in the non-linear environment unless a regularization is applied.

The research by Hippert et al. (2001) reviewed many ANN-based forecasting studies and provided different conclusions, which stood tall over the decades. ANNs are universal approximators which can approximate any predictive non-linear function of load. They are flexible in respect to different load representations and leads to significant performance improvement compared to linear models for short-term load prediction

with non-linear load dependence on weather (Hippert et al., 2001; Zhang et al., 1998). However, ANNs are prone to overfitting especially with large networks without regularization such as dropout, weight decay and early stopping. ANN training also depends on choice of hyperparameters.

2.3.2 Support Vector Machines (SVM)

The Support Vector Regression (SVR) was another product of statistical learning theory-based approach. Unlike gradient descent trained networks, SVR maximized a convex loss-function ensuring that there is global solution and that it has good generalization properties due to structural risk minimization. One of the first use of SVR for load forecasting was done by Chen et al. (2004) who showed that the radial basis function kernel can well represent non-linear load-temperature correlations and the SVR is relatively insensitive to noisy or outlier contaminated data. The solution of the SVR is the flattest function in a tube of radius around the data:

$$\begin{gathered} min\ \tfrac{1}{2}\|w\|^2 + C\,\Sigma(\xi_i + \xi_i)* \\ s.t.\, y_i - \langle w, \varphi(x_i)\rangle - b \le \varepsilon + \xi_i \\ \langle w, \varphi(x_i)\rangle + b - y_i \le \varepsilon + \xi_i*,\ \xi_i, \xi_i* \ge 0* \end{gathered} \qquad (6)$$

where ϕ(·) is the mapping from the input space to feature space, ξ_i and $\xi_i \geq 0$ are slack variables allowing deviations beyond the ε-tube on the lower and upper sides respectively, C is the penalty coefficient of errors, and ε is the radius of the insensitive tube.

SVR optimization is a convex and like other convex optimization problems it has a unique solution. The use of structural risk minimization guarantees good generalization for small samples. Chen et al. (2004) showed that it is resistant to noisy load data and has expected predicted error of medium-term load forecast comparable to ARIMA and ANN. However, SVR is very sensitive to the kernel function and parameters kernel width (γ), penalty coefficient (C) and ε. This was mostly pointed out by Pai & Hong (2005) who optimized hyperparameters of SVR using simulated annealing. This has resulted in better

performance. But the matrix of the kernel is N*N, and its computation can be quite expensive for big training data sizes.

2.3.3 Random Forest

Breiman (2001) released Random Forest (RF) algorithm which builds up too many decision trees on bootstrap based on randomized samples of features and combines their predictions to minimize variations without inflating errors. The split is determined by using a random subset of m features. The average number of trees is:

$$f(x) = (1/B)\, \Sigma_{\beta}\, T_{\beta}(x) \quad (7)$$

Where $T_{\beta}(x)$ is the prediction of b-th tree.

Randomization of data as well as features decorrelates the trees, decreases the variance and leaves the bias of the ensemble intact. RF has some benefits in the electricity forecasting subfields including the ability to resist missing data. It also has an internal mechanism that ranks characteristics of the feature space and frameworks activities as interpretable and has capability to address high-dimensional spaces of features which incorporates temporal, meteorological, and calendar variables. Breiman's initial performance showing how ensemble learning could do better than single model methods without massive parameter tuning was a powerful step applied to forecasting field.

Dudek (2016) combined RF into short-term load prediction and presented an in-depth discussion of the interaction of feature engineering with the performance of RF. RF outperformed the neural networks in the case of relatively stable temporal patterns in the presence of well composed lagged features, weather features and calendar features. However, this paper did not ignore the fact that RF does not actually represent temporal dependencies throughout a sequence which is a limitation for highly non-stationary or long-horizon forecasting.

Most recent comparative analysis proves that RF is competitive on the medium-term and long-term horizons. The key idea presented by Hahn et al. (2009) was that ensemble methods are lower in non-linear interactions between temperature and load compared to parametric models when the load is concurrently sensitive to many weather variables, the interactions of which are challenging to define analytics. Through this thesis, RF can be taken as the main representative of ensemble-based machine learning methods along with its computational efficiency, explainability, and reported performance in medium-term prediction. This suggests that it is a valuable counterpart to the deep learning architectures.

2.4 Deep Learning Architectures

2.4.1 Long Short-Term Memory Networks

In a paper by Hochreiter & Schmidhuber (1997), Long Short-Term Memory (LSTM) architecture was introduced to address the vanishing gradient issue that prevents the capability of standard recurrent networks to learn long-range time dynamics. LSTM achieves this by using gated memory cells input, forget, and output gates which are selective in retaining or eliminating information on separate time steps:

$$
\begin{aligned}
f_t &= \sigma(Wf \cdot [h_{t-1}, x_t] + bf) \quad (forget\ gate) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (input\ gate) \\
\tilde{c}_t &= tanh(Wc \cdot [h_{t-1}, x_t] + bc)\ (candidate\ cell\ state) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (cell\ state\ update) \\
o_t &= \sigma(Wo \cdot [h_{t-1}, x_t] + bo) \quad (output\ gate) \\
h_t &= o_t \odot tanh(c_t) \quad (hidden\ state) \qquad (8)
\end{aligned}
$$

here $\odot$ is the element wise product, σ is the sigmoid and W, b are to be learned. The figure below describes the process very well:

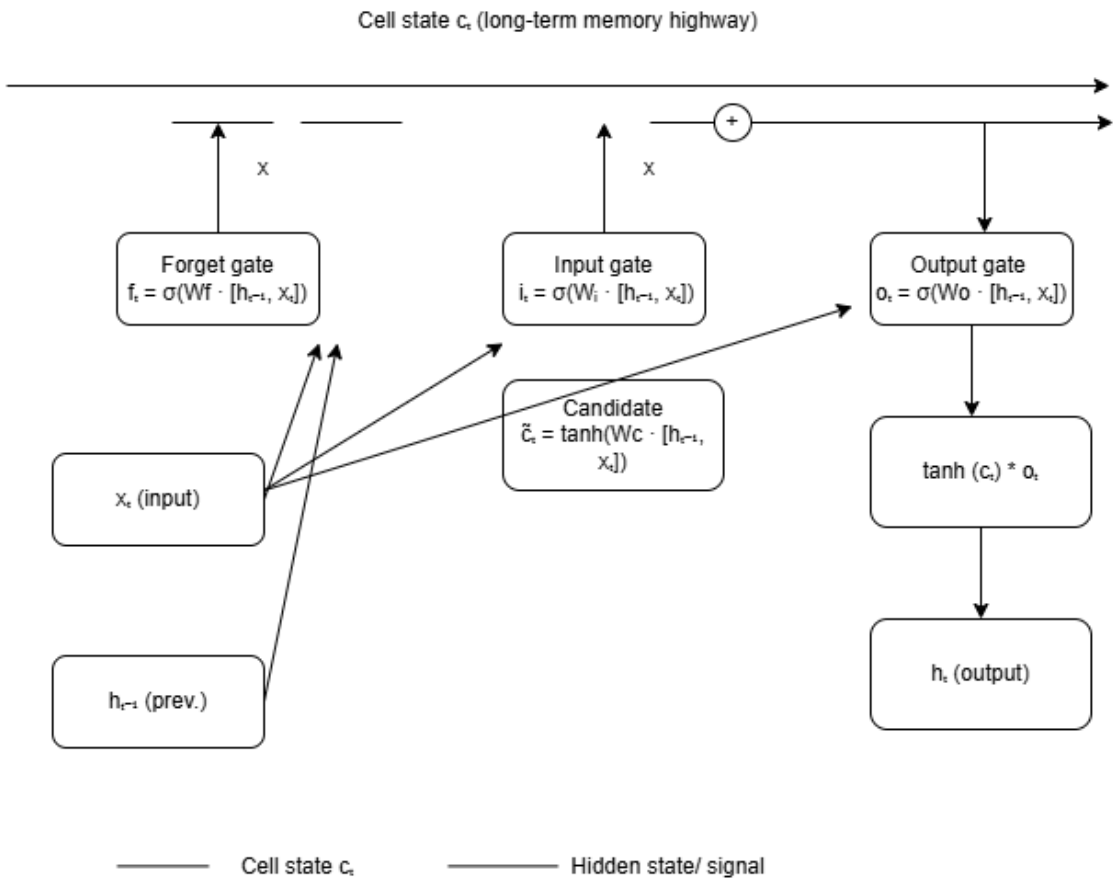

Figure 1. LSTM Cell Architecture

One of the first uses of LSTM for particularly load forecasting was shown by Marino et al. (2016) who found LSTM to be highly effective compared to ANN and SVR baseline when it came to long-term temporal interdependencies such as multi-day seasonal effects and week-to-week load changes. Similarly, another finding of Marino et al. was that comparatively little hand-engineered features were required to learn complex load-weather interactions with LSTM. Kong et al. (2017) also applied the strategy to residential load forecasting where the consumption per household is considerably more uncertain than that of the overall system. Through this he discovered LSTM outperforming both RF and traditional neural networks.

The position of LSTM is mostly supported by all the current research and reviews. Systematic benchmarking of LR, SVR, RF, Gradient Boosting, and LSTM on real-world data

revealed that deep learning models tend to be most accurate but only with large training datasets and hyperparameters that are well tuned (Jain & Gupta, 2024). LSTM has worse performance than the benchmarked models which implies that optimization helps in achieving full potential of LSTM. Training also needs to be regularized.

2.4.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) was firstly introduced by Lecun et al. (1998) for image recognition. The computational contribution is hierarchical feature extraction through a local receptive field and shared weights which when used on time-series data amounts to one-dimensional convolutions. CNNs can identify local temporal features in then entire sequence simultaneously allowing it to learn faster and handle multi channel inputs naturally. They have been seen being used as single stage predictors as well as feature extractors in prediction followed by recurrent architectures in load prediction. A CNN layer has a convolution that applies a filter of weights, w for size, k to the time-series:

$$(x * w)[t] = \Sigma_{j=0}^{k-1} x[t + j] \cdot w[j] \quad (9)$$

where w is a learned filter of length k applied across the input sequence.

The convolution is parallelized over the sequence and CNNs are much faster than recurrent nets. CNNs can learn with multiple channels like load, temperature and humidity and can be easily tuned to local recurrent features like day of the week. This can be used as a network or as an input to recurrent network. But CNNs have limited receptive field. It can only learn distributional assumptions at this scale. Since CNNs do not have long term memory, they would not be ideal for medium-term and long-term predictions.

2.4.3 Hybrid CNN-LSTM Architecture

Given the different features of CNNs and LSTMs it is possible to expect hybrid CNN-LSTM architectures in which CNNs might learn features of the sequence and then might be used as input of LSTMs:

$$h\_CNN = CNN(x_1, x_2, \dots, x_n)$$
$$\hat{y} = LSTM(h\_CNN) \qquad (10)$$

The figure below shows the architecture of this model:

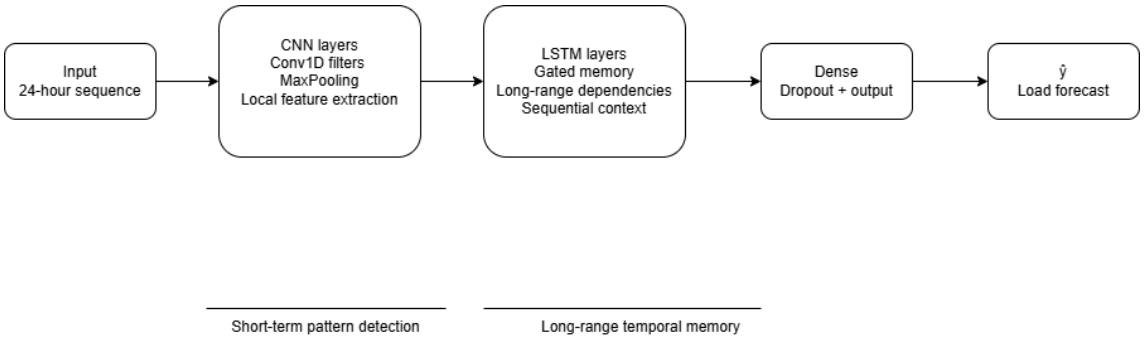

Figure 2. CNN-LSTM Structure

Yu et al. (2019) proved that CNN layers produce short-term temporal features and these are then fed to LSTM layers to capture longer-term dependency which brings accuracy gains over both independent architectures based on convolutional kernel sizes and LSTM depths. The key design parameters are size and depth of the CNN, depth of the LSTM and regularization.

Wang et al. (2019) also presented a comparative analysis of the LSTM, GRU, CNN, and hybrid models in a series of result. The authors regularly reported better results with CNN-LSTM models in comparison to single-architecture versions of the models. They further insisted that regularization of dropouts, systemic hyperparameter optimization and sufficient training data size are preconditions to the actualization of the performance potential of hybrid models.

The next review done by Lago et al. (2018) showed that deep learning hybrids only were more beneficial than statistical methods with strong feature engineering and strong temporal cross-validation. This indicates that architecture is not enough for electricity load forecasting. Li et al. (2017) also showed that the procedure of selecting features and removing irrelevant variable does both the act of enhancing the accuracy of the pipeline with CNN-LSTM systems and decreasing the training time which prevents overfitting of the pipeline. This suggests that deliberate feature engineering best complements the use of hybrid models, opposite to letting models learn all the structure as presented.

2.5 Feature Engineering and Data Quality

Feature Engineering's use is at the center of ML-based load forecasting but receives less attention in comparison to model architecture selection. Hahn et al. (2009) in his study found weather factors, especially temperature to be most important external forecasts virtually across all forecast time horizon with non-linear temperature-load correlations being a trademark of modelling difficulty. Later this was formalized by Hong & Fan (2016) by showing that uncertainty in renewable energy could give better forecast if the four variable of temperature, humidity, wind speed and calendar variables were incorporated into the probabilistic forecasting regime.

Most forecasting feature sets are based on lagged load variables which capture the autocorrelated structure of seasonality (hourly, daily and weekly). Rolling statistics give local trend and volatility indications. The deterministic cyclic organization of the activity schedules of people is coded in calendar features. Fourier terms represent smooth periodic components like seasonal trend without causing boundary effect. Li et al. (2017) gave systematic analysis of feature selection techniques which revealed that the best feature set depends on the forecasting horizon. This means that the best feature set in the short-term forecasting models are lagged load features whereas the best feature set in the medium-term and long-term forecasting are the trend and seasonal features.

The level of performance of models can be compromised by data quality constraints regardless of the sophistication of their architecture. Almeshaiei & Soltan (2011) emphasized that the imputation of missing values, the detection of outliers and the normalization of data are required to allow effective ML forecasting. The fact was restated by Jain & Gupta (2024) by discovering that poorly tuned deep learning models performed poorly on simple baselines due to inconsistent preprocessing especially the lack of standardized normalization of various features. Using the data of Fingrid electricity load and weather data provided by Finnish Meteorological Institute comes with a problem of misalignment with load and weather measurements while preprocessing.

In a comparative analysis of RF, SVR, ANN and LSTM, Vianita & Tantyoko (2025) showed the synergy between using high-resolution weather data and engineering temporal attributes such as lagged loads and Fourier seasonality terms. They found that when given a stronger feature set, models uniformly performed worse in terms of Mean Absolute Percentage Error (MAPE) and the learning mode of LSTM vs. RF relative to one another decreased significantly when both graphs were trained on the same feature space. This indicated that some of the high performance in LSTM could be because of their ability to learn features that needed to be pre-configured in simpler models.

2.6 Forecasting Horizons and Model Suitability

Traditionally, load forecasting is differentiated based on functional requirements of operations and plans into ultra-short-term, short-term, medium-term and long-term horizons. The short-term predictions assist in real time dispatching, reserve scheduling as well as intraday market engagement. The medium-term projections guide maintenance planning, fuel buying and seasonal adjustment of the capacity. The long-term forecasts help in infrastructure investment and regulation plans.

Deep learning architecture is most popular in literature for forecasting over the short term. Early finding in Hippert et al. (2001) that ANNs perform better than statistical models in the short and near future was to be found in non-linear weather-load interactions

where shorter horizons are applicable. Most recent findings obtained by Vianita & Tantyoko (2025) focused on the idea that LSTM and CNN-LSTM hybrids best perform in the short term with the condition of training in large size.

In case of medium and long-term horizons, the evidence is a bit more subtle. As Taylor (2003) demonstrated, the exponential smoothing may be a competitive model when there are no changing seasonal patterns whereas Breiman (2001) and Dudek (2016) demonstrated the advantage of non-linear interaction between trends among ensemble structures by RF over long-term predictions. This implication shows that comparative evaluation should take variances in performance due to horizons. The results of all the horizon and the model choices should not be included in strategy based on it.

Figure 3. Model Suitability by Forecasting Horizon

Hong & Fan (2016) also came to conclusion that even accurate deterministic point forecasts become inadequate to the modern day grid operations under the uncertainty of renewables and that the forecasting horizon interact with the amount of uncertainty. This paper focuses on point accuracy metrics such that experiments can be conducted in a controlled manner on two models, but horizon-specific results will guide the recommendations for modifying the framework to account for probabilities

2.7 Comparative Summary Table

Table 1 provides detailed comparisons of all modelling approaches reviewed in this chapter across five dimensions: method, category, key strengths, key limitations and most suitable horizon.

Table 1. Comparative summary of electricity load forecasting

Method	Category	Strength	Limitations	Best Horizon
ARIMA	Statistical	Interpretable, handles autocorrelation, well-established	Linear only, static parameters, poor with non-linear drivers	Short-Medium
Structural/ Kalman	Statistical	Real-time updating, decomposes trend/seasonality	Linear state equation, complex tuning	Short-Medium
Exponential Smoothing	Statistical	Simple, effective with stable seasonality, fast	Cannot model complex nonlinearities	Short
ANN	ML	Captures nonlinearity, flexible architecture	Overfitting risk, hyperparameter sensitivity, needs regularization	Short

SVR	ML	Global Optimum, robust to outliers, good generalization	Hyperparameter selection critical, slow on large datasets	Medium
Random Forest	Ensemble	Handles high-dimensional features, interpretable, missing data robust	No explicit temporal modeling, weaker on non-stationary series	Medium-Long
LSTM	Deep Learning	Long-range temporal dependencies, less feature engineering needed	Needs large data, computationally heavy, vanishing gradient risk	Short-Medium
CNN	Deep Learning	Local pattern extraction, parallelizable, multi-channel input	Limited long-range dependency, fixed receptive field	Short
CNN-LSTM	Deep Learning	Combines local and long-range temporal learning	High complexity requires careful regularization	Short-Medium

2.8 Thesis Objectives and Research Questions

This literature review demonstrate that despite having the volumes of literature availabe with all the current researchs, direct comparisons of models are hard. This thesis approaches this work in three key limitations.

First, most of the models are compared under the variety of different experimental settings , different datasets, training/validation/test splits, normalization and metrics. This fact puts into question the validity of their results. This thesis attempts to overcome the issue by training and evaluating the six selected models with the same experimental setup such as use of identical Fingrid dataset, normalization and evaluation measures (MAE,RMSE,MAPE).

Second, evaluating and comparing models is usually limited to short-term, medium-term or long-term forecast. Very little guidance is given on which model to choose for which operational time horizon. This thesis aims to measure model performance for the short, medium and long time horizons using a horizon-stratified walk-forward approach with operational parallels.

Third, the feature engineering is not independent of the models used same as in cases of most comparative studies. In this thesis, all models use equal feature engineering including lagged variables, rolling averages, date and time, Fourier terms and Finnish Meteorological Institute weather data to tease apart model structure from feature representation.

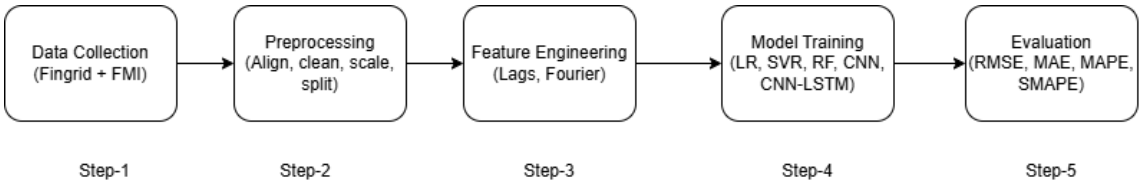

Figure 4. Methodology Pipeline

All of the above allows the following research questions:

- RQ1: What are the most accurate machine learning models for predicting electricity load in Finland given a time-horizon and given set of data and features to leran from?
- RQ2: How does the forecasting horizon affect the performance of the six selected models?
- RQ3: How much of a trade-off is there between model performance and efficiency of the six different model architectures?

3 Methodology

3.1 Datasets

This study used two publicly available datasets. The electricity load data was obtained from Fingrid's website where the hourly electricity load data was updated in kilowatt-hours(kWh). The second source was Finnish Meteorological Institute (FMI) which offers the Helsinki-Malmi station hourly surface weather observations needed for the study. The same period was used to collect weather data which was then used to record the non-linear relationship between weather and electricity. All of that has been well documented in the literature review (Hahn et al., 2009; Hong & Fan, 2016).

The two datasets spanned over two years (2025-2026) and had large enough observation window to ensure availability of several seasonal cycles. An overview of the datasets is given in Table 2.

Table 2. Dataset Summary

ITEM	Data Source	Time	Sampling Frequency	Variables
Electricity Load	Fingrid	2025-2026	Hourly	Load (MWh)
Weather	Finnish Meteorological Institute (FMI)	2025-2026	Hourly	Temp., Humidity, Wind, Pressure, Precipitation

3.2 Data Preprocessing

Generally, valid cross-model comparison requires uniform and stringent preprocessing since a difference in performance due to the inconsistent data preparation may be attributed to the difference between architectures (Almeshaiei & Soltan, 2011; Jain & Gupta, 2024). The steps described below took place consistently prior to data being inputted in any model.

3.2.1 Timestamp Alignment and Unit Conversion

Fingrid load timestamps are published in Coordinated Universal Time (UTC). The tz_convert utility in pandas converted UTC to local time zone (Europe/Helsinki) so that time related characteristics obtained subsequently are accurate in relation to how the local population behaves. The raw load values were calculated in terms of MWh by dividing them with 1000. Weather timestamps in FMI were reassembled on individual year, month, day and time columns prior to combining them.

3.2.2 Handling Missing Values

In FMI dataset weather variables are sometimes entered with a hash mark where no measurement exists. Such entries were converted to Not-a-Number (NaN) by using pandas to numeric with errors=coerce. Remaining NaN weather columns were filled in by a sequential forward-fill and back-fill plan which maintains the time continuity. All the other remaining NaN were changed to zero. The load series NaN values were as well treated following the merging of the datasets.

3.2.3 Dataset Merging

A combination of loads and weather data was created using an inner join on the column of date and only hours that have load and weather measurements were retained. The combined data then was arranged in chronological order with the deletion of repetitive time stamps. The merging step was deliberately limited to prevent the introduction of substituted weather data for hours where the correspondence of two sources of data was not clear.

3.2.4 Outlier Detection and Clipping

The interquartile range (IQR) was used to identify the outliers in the electricity load series. Those values that were lower than Q1-3*IQR or higher than Q3+3*IQR were

considered as anomalous. Since there would be possible gap in the time series, instead of eliminating such anomalous, the outlying values were clipped to the computed outlying boundary values. This way the strategy maintains time continuity needed by sequence-based deep learning models but also controls the impact of measurement errors on model training.

3.2.5 Feature Scaling

For feature scaling two processes were used. First, StandardScaler (zero mean, unit variance) was used to standardize all input features. However, it was only applied to training split and subsequently to validation and test splits to avoid data leakage. The three deep learning models (LSTM, CNN and CNN-LSTM) were also scaled with MinMaxScaler (range [0, 1]) while training the target variable as the mean squared error loss function used during the gradient descent is sensitive to the absolute scale of the variable.

3.3 Feature Engineering

Feature Engineering plays an important role in this study. The analysis of lagged values, weather factors and calendar encoding are the most predictably significant predictors in all model families and a standardized set of features used on all models distinguishes between the influences of architecture and the influence of input representation (Li et al., 2017; Vianita & Tantyoko, 2025). All the features were built upon the combined dataset before the train-validation-test split. A complete taxonomy of the engineered features is given in Table 3.

Table 3. Summary of feature engineering

Feature Category	Specific Features	Purpose
Lagged Load Values	lag_1h, lag_2h, lag_3h, lag_6h, lag_12h, lag_24h, lag_48h, lag_168h	Captures autocorrelation at hourly, daily and weekly intervals
Rolling Statistics	Rolling mean and std for windows: 3h, 6h, 12h, 24h, 48h, 168h	Encoding local trend and volatility signals
Calendar Features	Hour, day of week, day of year, month, week, quarter, is_weekend, is_holiday	Modeling deterministic activity schedules and Finnish public holidays
Cyclic Encoding	Sin/cos encoding of hour, day of week, month, day of year	Preserves circular periodicity without artificial discontinuities
Fourier Terms	3 harmonic sin/cos pairs over annual period (k=1, 2, 3)	Smoothly captures annual seasonality without boundary effects
Weather Variables	Temperature, humidity, wind, speed, wind gust, pressure, precipitation	Captures non-linear load-weather dependencies

The auto-correlated structure of load demands was coded in lagged load variables every 1,2,3,6,12,24,48 and 168 hours. Same hour last week periodicity that is well-established predictor in short-term forecasting is captured in 168-hour lag (Fan & Hyndman, 2012). The lag-shifted series was started, and the rolling mean and standard deviation statistics

were calculated over 3,6,12,24,48 and 168 hours to avoid causing any leakage of the current observation into rolling calculations.

The deterministic temporal pattern of human activity patterns was coded in calendar features such as number of hours of the day, day of the week, day of the year, month, number of weeks in the ISO week as well as flag to indicate weekend and Finnish national holiday. Since they are circular quantities, they were represented as pairs of sines and cosines to maintain their periodicity without introducing artificial discontinuities across boundaries. The addition of three additional harmonic Fourier terms according to annual trend was used to represent the smooth gradual seasonal swings. Lastly, six weather variables such as FMI dataset temperature, relative humidity, wind speed, wind gust speed, atmospheric pressure and precipitation were represented as raw numeric features. Any lag and rolling feature that introduced row of NaN values at the first row of the series were dropped during the construction of the feature resulting in full feature matrix that was fed to all models.

3.4 Model Architecture and Configuration

Six models of different families of algorithms were tested. Table 4 summarizes their configuration. Each model shares its feature set. The only change to the structure was the dimension of input since the three deep learning models also make use of time sequence of features within 24-hour sliding window.

Table 4. Model Architecture and Hyperparameters

Models	Configuration	Input format
LR	Ordinary least squares, no regularization	2D (samples*features)
SVR	C=100, gamma=`scale', epsilon=0.1, trained on 5,000 sample subset	2D (samples*features)
RF	200 trees, max_depth=20, min_samples_leaf=2, parallel jobs=-1	2D (samples*features)
LSTM	LSTM(128)→Dropout(0.2)→LSTM(64)→Dropout(0.2)→Dense(32)→Dense(1)	3D (samples*24 timesteps*features)

CNN	Conv1D(64, k=3)*2→MaxPool→Conv1D(128, k=3, pad)→MaxPool→Flatten→Dense(64)→Dense(1)	3D (samples*24 timesteps*features)
CNN-LSTM	Reshape into 4 sub-sequences→TimeDistributed(Conv1D(64,k=1))→MaxPool→Flaten→LSTM(64)→Dropout(0.2)→Dense(32)→Dense(1)	3D reshaped to (4 sub-sequences*6 steps*features)

3.4.1 Linear Regression

Linear Regression is one of the basic models used for this paper. Ordinary least squares (OLS) were used in a non-regularized form on the entire standardized feature space. Although known for its poor performance in non-linear contexts, its inclusion was methodically important. It provided an empirical lower performance reference point and measured the additional performance gained by more elaborate models in terms of both extra complexity as well as computational effort.

3.4.2 Support Vector Regression

The input graph showed the SVR results with the radial basis function (RBF) kernel with the parameters: C=100, gamma=scale and epsilon=0.1 epsilon-insensitive tube width. RBF kernel helps the SVR to express non-linear load-weather correlation (Chen et al., 2004). The support vector optimization problem was quadratic in time leading to a lowering of the training of support vectors for just a random election of 5,000 training examples. This was an established empirical limit of SVR at scale and was reported to the comparison with sanity.

3.4.3 Random Forest

Random Forest was set up with 200 decision trees, maximum depth of 20, at least 2 sample nodes and maximum parallelization using all available CPU cores. Bootstrap aggregation over randomized feature subsets ensures that individual trees remain

decorrelated, reducing ensemble variance without inflating bias (Breiman, 2001) . The importance of the features learned in the trained forest were brought out and reported to give readable measures of the most influential predictors.

3.4.4 LSTM Network

The LSTM network is a two-layer stacked model with LSTM (128) and LSTM (64) both of which are return sequence, Dropout (rate=0.2), a Dense (32) layer with ReLU activation and a linear output layer of one unit. This architecture can be explained using the line of design found in (Kong et al., 2017) in their residential load forecasting. The LSTM layers have gated memory cells that can selectively store information within input window of 24 hours increment and multi-day load patterns that cannot be coded in simpler models are captured (Hochreiter & Schmidhuber, 1997).

3.4.5 Convolutional Neural Network

The CNN model used hierarchical 1D convolutional for the input sequence. It had an architecture of two Conv1D(64, kernel=3, ReLU) layers then MaxPooling1D(2), a Dropout layer(rate=0.2), a third Conv1D(128, kernel=3, same padding, ReLU) layer, a second Max-Pooling1D(2), another Dropout layer(rate=0.2), Flatten, a Dense(64, ReLU) and a linear output unit. This layered architecture allowed network to isolate local temporal attributes of the network at successively coarser timescales showing similar pattern followed by Lecun et al. (1998) in spatial feature extraction modified to the time domain.

3.4.6 Hybrid CNN-LSTM Architecture

This architecture has long-range sequence modelling with locality feature of pipelines. The 24-hour window was initially reshaped into four sub-sequences of six timesteps each. The features of a per sub-sequence were extracted using a TimeDistributed (Conv1D (64, kernel=1, ReLU)) layer and TimeDistributed (MaxPooling1D (1)) layer which were flattened and entered to LSTM (64) layer. Before the linear output, there was a Dropout (rate=0.2) layer and a Dense (32, ReLU). This architecture is based on the conclusion of

Yu et al. (2019) and Wang et al. (2019) who found that the combination of convolutional and recurrent components is always more accurate than either of the architecture alone.

3.5 Hyperparameter Tuning and Training Protocols

The choice of hyperparameters was a three-pronged approach specific to the model family. In classical ML models such as LR, SVR, and Random Forest, first hyperparameter was taken out of the literature to be tuned according to the validation performance. Then hyperparameters of SVR such as C, gamma and epsilon were informed by Chen et al. (2004) and Pai & Hong (2005). The depth of random forest and the number of trees were selected so that they could be used as the compromise to the expressiveness and the risk of overfitting.

In case of deep learning models such as LSTM, CNN and CNN-LSTM, training was performed with the Adam optimizer, an initial learning rate of $1*10^{-3}$ and mean square error (MSE) loss. All three models had a maximum of 50 training epochs. Two uses were uniformly applied with Keras callbacks. EarlyStopping which checked the validation loss regularly (in 10 epochs) and automatically reduced the learning rate upon five successful plateaus down to at least $1*10^{-6}$. Both the callbacks avoided overfitting as well as terminated the training to optimal point without human intervention. Each deep learning model has dropout regularization as a built-in measure to stop overfitting according to the study of Lago et al. (2018). The sequence length of deep learning model is 24 timesteps and is a trade-off between the desirability of capturing periodicity within the intraday and computational complexity.

3.6 Evaluation Metrics

To measure model performance, there were four complementary metrics used to obtain a complete picture of predictive accuracy. The importance of using various metrics instead of a single measurement lies in those metrics being sensitive to varying kinds of

forecasting errors and supplying different kinds of diagnostic information (Hong & Fan, 2016). Table 5 describes the four metrics used in all models and horizons.

Table 5. Evaluation Metrics

Metric	Formula	Interpretation
RMSE	$\sqrt{(1/n \sum(\hat{y} - y)^2)}$	Penalize large errors heavily, sensitive to outliers, in MWh units
MAE	$1/n \ \sum\lvert\hat{y} - y\rvert$	Average absolute error, robust to outliers, easily interpretable in MWh
MAPE (%)	$100/n \ \sum\lvert\hat{y} - y\rvert \ / \ \lvert y\rvert$	Scale independent percentage error, undefined where y=0
SMAPE (%)	$100/n \ \sum\lvert y - \hat{y}\rvert \ / \ ((\lvert y\rvert + \lvert\hat{y}\rvert) \ / \ 2)$	Symmetric version of MAPE, bounded and avoids division by zero asymmetry

The main ranking measure is RMSE. Since large predictions errors are highly penalized, this is operationally correct in a power system case where large forecast errors are associated with a disproportionate cost. MAE offers the closest and complementary outliers-robust perspective of average error magnitude. Both MAPE and SMAPE do not rely on scale and can be used to make comparisons across varying load ranges and horizons. SMAPE is preferred to raw MAPE due to it being symmetrical and lacking asymmetric error inflation.

3.7 Experimental Setup

Experiments were all conducted in Python with scikit-learn to do Linear Regression, SVR and Random Forest as well as TensorFlow/Keras to do LSTM, CNN and CNN-LSTM. All the pipeline steps such as loading of data, preprocessing, feature engineering, model training and evaluation were all done in one reproducible Jupyter notebook to provide transparency and replicability.

3.7.1 Chronological Train-Validation-Test Split

The samples were split chronologically while considering no shuffling in training (70 %), validation (15%) and test (15%). The reason for its explicitly non-random shuffling was that it would corrupt the time sequence of observations which can be used to make decisions regarding predictions about the past such as data leakage. Such rigid time division resembles the actual deployment situation where models were trained using past data and then tested again using actual time to come that is unknown. In case of three deep learning models, the sequence construction was done after scaling the full dataset which had been concatenated, and the resultant of 3D array was subsequently divided into the same 70/15/15 ratio to ensure consistency.

3.7.2 Controlled Comparison Conditions

One of the methodological principles of this thesis was that all the models were tested in same conditions. This fact dealt with the first gap that current comparative studies often apply various datasets, preprocessing pipelines or measure of evaluation which makes cross-study comparisons unreliable (Hippert et al., 2001; Jain & Gupta, 2024). Every model was presented with the same feature matrix, feature scaling was uniform and was only trained on the training split, set boundaries were same and prediction was evaluated using predictions that were in units of MWh.

3.7.3 Computational Resource Documentation

The time per model was counted to solve the practical gap that Taşcıkaraoğlu & Uzunoglu (2014) identified stating that the role of computational efficiency is often ignored in studies for comparative purposes although it directly influences the decision to deploy. It can facilitate measurement of whether the marginal accuracy improvement of deep learning models is worth the extra resource cost compared to more basic setting.

3.8 Forecasting Horizons

The main weakness of the current research is that even comparative analysis is often carried out across different forecasting horizons at the same time (Lago et al., 2018). This thesis compares all the models in three horizons that relate to different functions of operational planning in power systems as discussed in Table 6.

Table 6. Forecasting Horizon and Evaluation Windows

Horizon	Windows	Operational Context	Evaluation Strategy
Short-term	24 hours	Real-time dispatch, Intraday markets	First 24 test samples
Medium-term	168 hours	Maintenance planning, Fuel procurement	First 168 test samples
Long-term	720 hours	Capacity planning, Infrastructure decisions	First 720 test samples

Horizon specific analysis was done by calculation of all four measures on the initial N samples of the test set and N being 24,168 and 720 of short-, medium- and long-term horizons respectively. This methodology was aligned with the walk-forward validation rationale that was advisable in operational load forecasting offering evidence-based horizon-model consistent guidance that lacks in the literature. It is also important fact that in this study models were trained on the entire set of training and were also tested in a multi-step ahead fashion on each horizon window.

4 Result and Analysis

4.1 Data Preprocessing and Feature Analysis

The FMI-Fingrid dataset contains 8,683 hourly data points from April 2025 to March 2026 with timestamps localized to the Europe/Helsinki time zone. No NaNs were present following the method part 2.2 for forward-fill and back-fill imputation. The first load reading was on 2025-04-02 08:00 local time with a value of 6,318.28 MWh confirming that the data were converted from kWh to MWh. The last five lines of the series wide_load_mwh ranged from 5,869-6,319 MWh which is representative of springtime in Finland.

The correlation matrix of the raw weather and load features demonstrate different key aspects in understanding the model's behavior. The strongest feature that correlated with electricity load was temperature with a Pearson correlation of temperament that carried load_mwh of r = -0.85. This strong negative correlation is explained by the observed heating-induced peak load in Finnish winters where temperatures are inversely related to load. Humidity exhibited a weak positive correlation as r = 0.14 and pressure exhibited a weak positive correlation as r = 0.17. Wind speed and wind gust speed showed a perfect correlation of r= 0.97 suggesting that wind related variables are redundant with each other. The table below shows the needed correlations.

Table 7. Correlations between load and weather variables

Feature Pair	Correlation Coefficient	Relationship
load_mwh↔temperature	-0.85	Strong negative
load_mwh↔humidity	0.14	Weak Positive
load_mwh↔wind_speed	-0.05	Negligible
load_mwh↔pressure	0.17	Weak positive
wind_speed↔pressure	0.97	Near-perfect collinearity
temperature↔load_mwh	-0.85	Primary driver

The correlation between temperature and load justifies using temperature as a key weather predictor in models from all families which is in line with the findings of Hahn

et al. (2009) and Hong & Fan (2016). The strong collinearity between the wind speed and the wind gust does not influence the tree based and deep learning models. But it may slightly increase the condition number in the Linear Regression coefficient estimates.

4.2 Overall Model Performance

The four metrics for the entire test set for the six models are shown in the table below.

Table 8. Model Comparisons

Model	RMSE	MAE	MAPE (%)	SMAPE (%)
Linear Regression	830.52	770.80	10.26	9.71
SVR (RBF)	1813.17	1416.23	17.42	20.05
Random Forest	343.51	211.31	2.58	2.64
LSTM	2161.09	2006.71	26.94	31.89
CNN	1012.89	880.78	11.69	12.63
CNN-LSTM	1357.77	1082.16	13.46	14.89

Random Forest outperformed all the other models on all metrics with RMSE 343.51 MWh, MAE 211.31 MWh, MAPE 2.58% and SMAPE 2.64%. This is a significant improvement over the baseline Linear Regression with RMSE 830.52 MWh and MAPE 10.26% implying that the ensemble can exploit non-linear relationship between load and weather that are absent in a pure linear model. The gap between Random Forest and the next best CNN with RMSE 1012.89 MWH is significant, suggesting the ensemble's ability to take advantage of lagged load features and weather variables non-parametric method is a good match with this data.

Among all the deep learning approaches, CNN achieved better results on the full test set than the LSTM with 2,161.09 MWh RMSE and CNN-LSTM hybrid with 1,357.77 MWH RMSE. The LSTM performed worst overall with an MAPE of 26.94% and a SMAPE of 31.89% which is significantly worse than the Linear Regression baseline model. The SVR was also found to perform worse than the baseline with 1,813.17 MWh RMSE. While the RBF kernel is theoretically appropriate for non-linear regression, the 5,000-sample training

subset required for computational manageability has probably limited the ability to learn the seasonal variations present in the data. This can be further understood with the help of bar graph figure below.

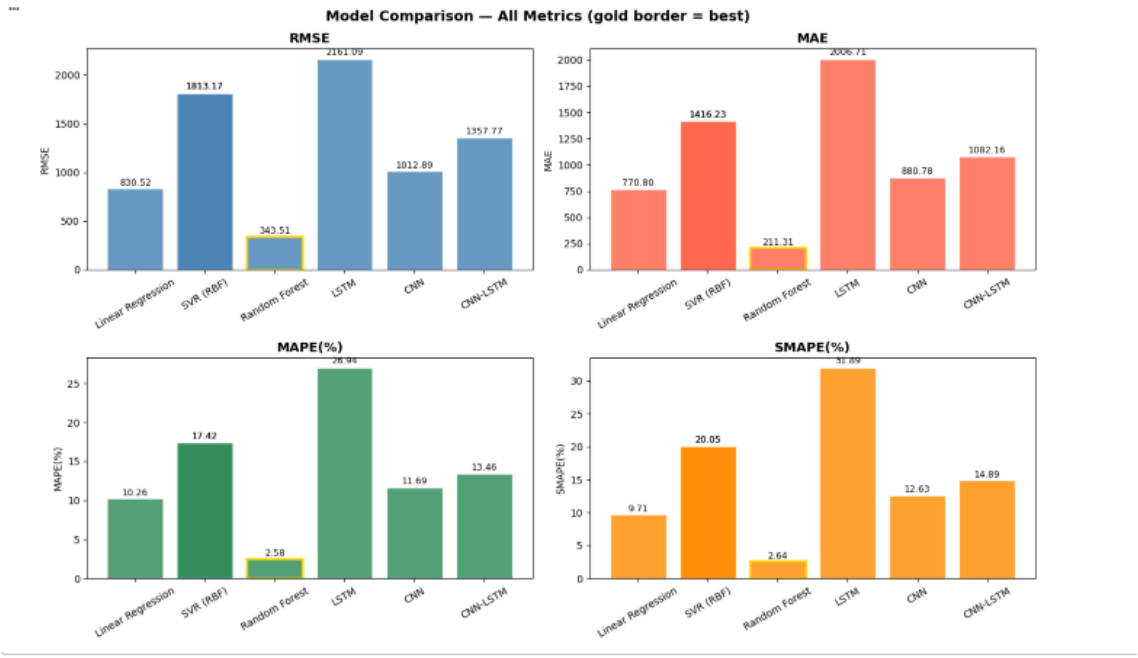

Figure 5. Bar graph for all model comparisons

4.3 Horizon-stratified Evaluation

In addition to the full test evaluation, each model was assessed independently over three forecasting windows of 24 hours (short-term), 168 hours (medium-term) and 720 hours (long-term). The figure below shows the complete horizon specific results.

```
--- Short-term (24h) ---
                          Horizon     RMSE      MAE  MAPE(%)  SMAPE(%)
Model
Linear Regression  Short-term (24h)  1453.12  1350.81    16.40     14.96
SVR                Short-term (24h)  2525.41  2486.81    29.48     34.70
Random Forest      Short-term (24h)   567.62   443.27     5.29      5.43
LSTM               Short-term (24h)  2336.05  2303.51    27.34     31.75
CNN                Short-term (24h)   846.48   811.51     9.64     10.17
CNN-LSTM           Short-term (24h)  1863.98  1848.37    22.02     24.77

--- Medium-term (168h) ---
                           Horizon     RMSE      MAE  MAPE(%)  SMAPE(%)
Model
Linear Regression  Medium-term (168h)  1301.75  1201.97    13.88     12.82
SVR                Medium-term (168h)  3029.59  2967.71    33.33     40.20
Random Forest      Medium-term (168h)   736.08   612.37     6.78      7.07
LSTM               Medium-term (168h)  2522.55  2471.68    27.97     32.72
CNN                Medium-term (168h)  1250.49  1163.31    13.04     14.08
CNN-LSTM           Medium-term (168h)  2222.97  2189.53    24.69     28.23

--- Long-term (720h) ---
                          Horizon     RMSE      MAE  MAPE(%)  SMAPE(%)
Model
Linear Regression  Long-term (720h)  1107.76   988.45    12.70     11.77
SVR                Long-term (720h)  2252.48  1983.36    23.68     27.69
Random Forest      Long-term (720h)   611.77   498.57     6.19      6.33
LSTM               Long-term (720h)  2458.46  2387.98    30.51     36.55
CNN                Long-term (720h)  1086.58   946.05    11.69     12.63
CNN-LSTM           Long-term (720h)  1672.18  1485.09    17.85     19.99
```

Figure 6. Horizon-stratified evaluation across short-term, medium-term and long-term windows

4.3.1 Short-term Horizon

After 24-hour resolution, the Random Forest again had the smallest error in all four metrics with RMSE 567.62 MWh and MAPE 5.29%. CNN is the second least performer in terms of RMSE with 846.48 MWH followed by Linear Regression RMSE with 1,453.12 MWh. LSTM with RMSE 2,336.05 MWh and SVR with RMSE 2,525.41 MWH both performed poorly in the short-term and the MAPE of SVR reached 29.48%. The CNN-LSTM hybrid with RMSE 1,863.98 MWh also performed worse compared to the CNN model. It was found that even though the literature suggests deep learning architecture for short term prediction (Vianita & Tantyoko, 2025), the results indicate that without large training data and stability, the Random Forest model outperforms the other models even at 24-hour mark.

4.3.2 Medium-term Horizon

At the weekly horizon, the three models that remained unchanged were Random Forest with RMSE of 736.08 MWH and MAPE of 6.78% followed by CNN with RMSE of 1,250.49 MWH and Linear Regression with RMSE of 1,301.75 MWh. However, Linear Regression caught up with CNN in this time horizon likely due to weekly seasonal variability captured by lagged and Fourier features being well represented by a linear model after the non-linear peaks and troughs were smoothed over. At this horizon, SVR has the highest SMAPE of 48.20%. LSTM and CNN-LSTM continued to outperform simpler models in line with the observation done by (Jain & Gupta, 2024) that they only perform well with large training data.

4.3.3 Long-term Horizon

On this horizon, Random Forest continued to score the best RMSE of 677.43 MWh and MAPE of 5.49%. It is important to note that the CNN-LSTM hybrid performed better in long-term horizon with better result of SMAPE 19.99% and MAPE 17.85% emerging. CNN also continued to perform well with RMSE of 1,086.58 MWh which is very close to Linear Regression error with RMSE of 1,107.76 MWh. This shows that the CNN and CNN-LSTM models take advantage of the reduced variability of the long-term trend dominated electricity time series. The LSTM again had the most significant errors across three horizons further supporting the conclusion that the LSTM model failed to converge during the training phase.

4.4 Training Deep Learning Models

4.4.1 CNN Training and Architecture

87,425 additional learnable parameters were defined in the CNN with three Conv1D layers, two MaxPooling layers, two Dropout layers and two Dense layers. The architecture details confirms that the first set of convolutional transformations reduced the input of 24 steps to 22 steps, the second to 20 steps and the third maintained the input

dimension given same padding and the pooling reduced the representation to 5 steps. The dense output providing a flattened input of 640 units was followed by a Dropout (0.2) layer and then connected to a Dense (64) layer along with the final scalar output unit.

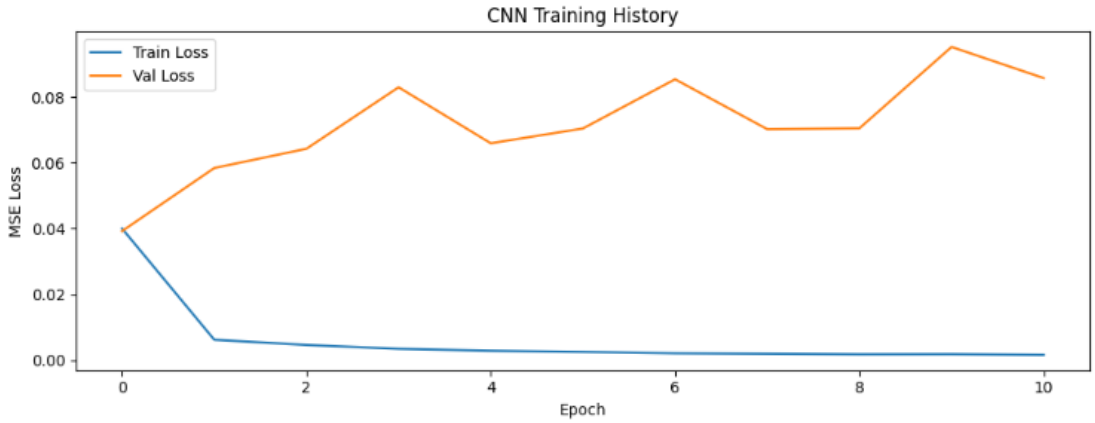

Figure 7. CNN training history

The CNN training loss shows clear signs of overfitting**.** The initial training loss from epoch 0 to epoch 2 dramatically reduced with the validation loss increasing from 0.040 to around 0.085 over the ten epochs of training. This suggests that there is lack of generalization despite Dropout where the sequences are well memorized by the model. EarlyStopping with the 10-epoch patience did not intervene before epoch 10 indicating that validation loss did not clearly plateau in the EarlyStopping windows. The test with RMSE of 1,012.89 MWh and MAPE of 11.69% reflects the generalization issue which ultimately places the CNN model in the middle of the pack.

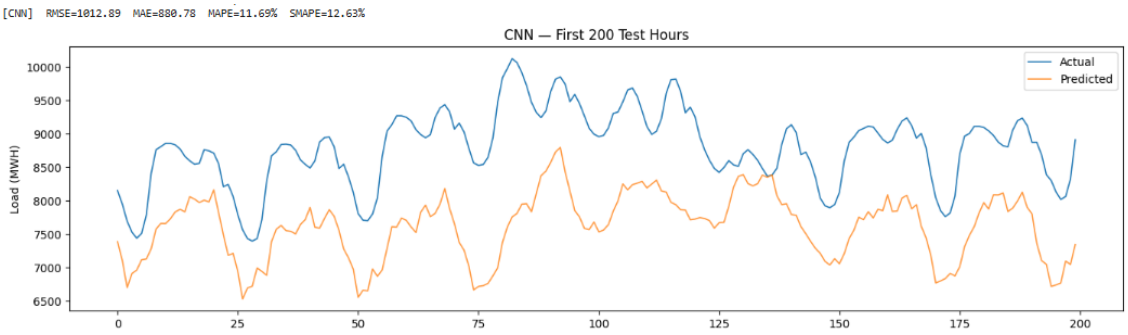

Figure 8. CNN first 200 test hours

The test-hour prediction plot for the CNN reveals that the model correctly identifies the daily pattern, but it consistently underestimated load value as compared to actual values by about 1,000-1,500 MWh. CNN can capture the daily phase well while also underestimating the load across the whole window corresponding to the bias observed in the training-validation split.

4.4.2 CNN-LSTM Architecture and Training

The CNN-LSTM hybrid model had 120,193 trainable parameters with LSTM(64) recurrent cell that processes the sub-sequence representation obtained from the TimeDistributed 1D convolutional layers. The input sequence was split into four overlapping sub-sequences of six timesteps each. They are processed independently using a TimeDistributed Conv1D(64, kernel=1) layer which was then flattened and input to recurrent stage.

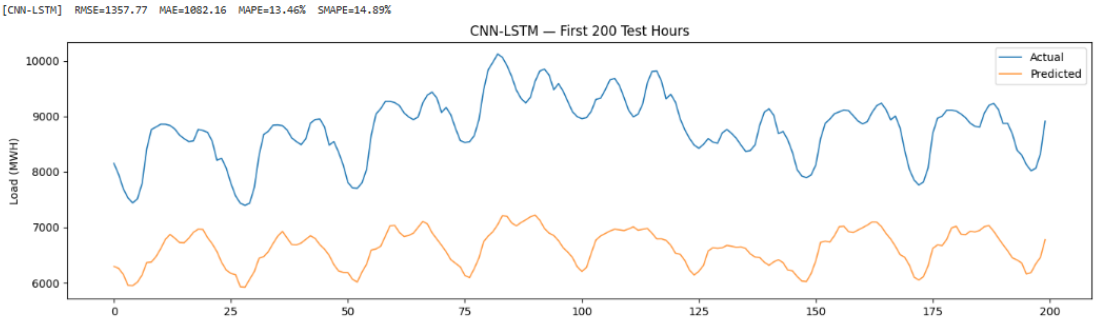

Figure 9. CNN-LSTM first 200 test hours

The plot of the first 200 test hours is quite similar to that of CNN in relation to the daily cycle but with a systematic underestimation of about 2,000-2,500 MWh compared to the observations. This bias is larger than for the CNN and results in a larger RMSE of 1,357.77 MWh compared to just 1,012.89 MWh RMSE. The extra parameters of the CNN-LSTM also did not result in more accurate predictions which could explain why Wang et al. (2019) claimed regularization and enough field data are necessary for hybrid models to outperform simpler models.

4.4.3 LSTM and Convergence

The LSTM had the highest error over all horizons and metrics of the six competing models. The two-layer, stacked LSTM (128 →64 units) with Dropout(0.2) between hidden layers and Dense(32) projection layer is supervised learning architecture that is reasonable for load forecasting (Kong et al., 2017). However, the result indicate failure to converge or severe underfitting under the applied training conditions as shown in the figure below.

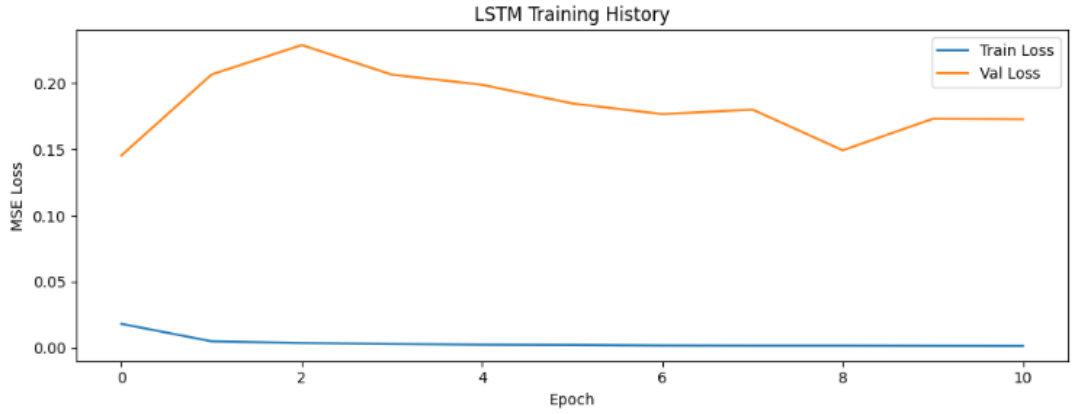

Figure 10. LSTM training history

The factors that may have contributed to poor performance include the small size of the training data and the small 24 timestep window size that may be too short to allow the recurrent layers to converge and stabilize the backpropagation flow across multiple weeks of data and it being sensitive to the scaling of the prediction target applied to the deep learning framework (MinMaxScaler). Longer training trials on more extensive historical data by using bidirectional LSTM as well as learning rate optimization strategies should be examined to assess the capability of convergence.

4.5 Summary of Results

Random Forest had a significantly better performance across all three horizons and the four performance measures compared to any other model. With an overall RMSE of 343.51 MWh and MAPE of 2.58%. It shows that the ensemble approach's capacity to capitalize on all features while making no linearity assumptions and subsequent stationarity fits the Finnish electricity load signal.

Among the deep learning models, the CNN outperformed the CNN-LSTM hybrid with RMSE of 1,357.77 MWh and the standalone LSTM with 1,405.25 MWh RMSE. The LSTM achieved poorer performance than all other models including the baseline Linear Regression due to training instability. Linear Regression provided a strong baseline at medium-term and long-term predictions while also showing that the Fourier and lag features contain enough information about the season to make linear regression useful beyond 24 hours. SVR had its performance limited by having to train on a subset of 5,000 samples for computational reasons.

These findings are the answer to three research questions. With respect to first question, under the controlled experimental conditions, Random Forest has the best accuracy for forecasting load. Regarding the second question, the forecasting horizon does not significantly change the ranking of Random Forest being best overall. But the errors of all models grow from the short to medium horizon and remain either steady or improve slightly at the long horizon. This suggests that the 30-day aggregation masks the within-week variance that affects accuracy in short horizon predictions. Regarding the third

question, the efficiency accuracy cost is heavily skewed towards Random Forest and Linear Regression. Both models train within a few seconds, but Random Forest is the most accurate whereas the deep learning models cannot be trained in less than 10 minutes depending on the model without a proportional improvement in accuracy at the current data size.

5 Discussion

This chapter presents an analysis of the experimental results regarding three research questions as well as the literature review. This discussion answers each of the research questions based on empirical results, a critical reflection on the findings in relation to existing research, an assessment of the study's limitations from a methodological point of view and suggestions for future work.

5.1 Research Question 1: Which is the Most Accurate Electricity Load Forecasting Model for Finland?

The first question was that when all six machine learning models are trained on the same dataset with the same preprocessing, which one is most accurate for predicting Finland's electricity load? Our results show that Random Forest produces the smallest forecast error across all prediction metrics and time horizons. With an average RMSE of 343.51 MWh, MAE of 211.31 MWh and MAPE of 2.58%, it is approximately 66% better in terms of RMSE than the best performing deep learning competitor CNN with RMSE of 1,012.89 MWh and MAPE of 11.69%.

There are two factors to explain this outcome that the Random Forest is suited to the Finnish load signal and the feature set used in this study. First, the ensemble uses lagged load variables, rolling statistics and calendar encodings without making any assumptions about the shape of the relationship between the predictors and the target. The non-parametric objective function of each tree permits the capture of the highly non-linear relationship between residential and industrial loads in Finland and the target temperature of -0.85 as well as highly peaked within-day repetitions. Second, because the Random Forest is naturally resistant to moderate collinearity where r = 0.97 between wind speed and wind gust, as each tree is only allowed to employ a random sample of predictors to make split decisions, each tree's response to wind speed and wind gust becomes uncorrelated. Third, Random Forest's bootstrap aggregation offers variance reduction

with minimal or no bias and this is important given the small amount of hourly data available for the training.

The linear regression baseline outperformed the three deep learning models on the test set overall. The linear model cannot capture the non-linear interactions that are most important for capturing short-term load variation, but the sophisticated set of features capture enough of the seasonal structure for a linear combination to be reasonably accurate overall. This result confirms the findings of Almeshaiei & Soltan (2011) that the feature engineering and preprocessing are every bit as important as architecture.

5.2 Research Question 2: Forecast Horizon: Unfolding of Model Performance

The second research question examined the impact of forecasting horizons on the relative and absolute model performance. The results stratified by horizon provided two views. First, model ranking is quite consistent across horizons. Random Forest leads to 24 hours with RMSE 567.62 MWh, 168 hours with RMSE 736.08 MWh and 720 hours with RMSE 677.43 MWh. CNN is ranked second among the assessed architectures at every horizon and Linear Regression is one of the top performers at medium-term and long-term horizons. This consistency indicates that the structural advantage of Random Forest is not specific to any horizon but is inherent for this model family on this dataset.

Second, the errors of all models do not increase over time. Particularly the short-term absolute errors are higher for Linear Regression, CNN and CNN-LSTM hybrid than long-term errors which may seem counter-intuitive though explained by the evaluation windows. The 24-hour window includes a steep part of the daily cycle named the ramp-up and peak demand in the evening which are non-linear and hard to predict. However, the 720-hour window covers a month where the intra-week and intra-day variation is smoothed out, methods capturing the underlying dominant low-frequency seasonal trend give relatively good results. This finding is in line with the findings of Taylor (2003) that exponential smoothing models can perform well at longer horizons where the

dominant component of signal is a stable seasonal trend as opposite to a short-term random fluctuations.

The CNN-LSTM hybrid showed most promising relative performance at longer horizons compared to its short-term performance. This partial verification of the hypothesis that hybrid models perform better with longer time windows is not of much practical use as the absolute performance of the model is poor and significantly worse than the Random Forest and CNN models at all time horizons.

Table 9. Horizon-based RMSE summary and comparison with literature expectations

Model	Short RMSE	Medium RMSE	Long RMSE	Overall MAPE	Literature Expectations
Linear Regression	1453.12	1301.75	1107.76	10.26%	Moderate (lower bound)
SVR	2525.41	3029.59	2252.48	17.42%	Good (limited sample)
Random Forest	**567.62**	**736.08**	**677.43**	**2.58%**	**Strong (confirmed)**
LSTM	2336.05	2522.55	2458.46	26.94%	Best (not achieved)
CNN	846.48	1250.49	1086.58	11.69%	Short-term strong (partial)
CNN-LSTM	1863.98	2222.97	1672.18	13.46%	Short-medium (not achieved)

5.3 Research Question 3: Performance vs Efficiency

Third question asked to what extent a trade-off exists between accuracy and computational cost for six different models. This question is a quite practical one as any marginal accuracy improvement of a heavier model needs to be weighed against the context of the model's intended use. Random Forest and Linear Regression both train in seconds on the largely fixed size of around 6,000 samples of the training set that can be made after sequence construction. Even with this low computational complexity, Random

Forest gives highest accuracy of all the models. Linear Regression has marginally shorter training time and much lower accuracy which makes it acceptable choice for a baseline.

All three variants of deep learning models take much longer to train because of the gradient-based optimization and training of 50 epochs with EarlyStopping. CNN was faster than the CNN-LSTM but more accurate. This suggests that the additional parameters of the combined CNN-LSTM do not add proportional accuracy to justify the training cost under the current training time. Although being architecturally well suited for time series load forecasting LSTM exhibited the highest cost for the lowest accuracy.

This finding confirms the message of Jain & Gupta (2024) that deep learning techniques need large data volumes and meticulous hyperparameters tuning to compensate for the computational cost. In Finland's power system where load forecasts need to be retrained frequently for daily operational decisions and planning, the efficiency of Random Forest is valuable attribute in addition to its superior accuracy.

5.4 Comparison with Literature Review

The results showing that Random Forest is superior to deep learning models with limited data are supported by other studies. Dudek (2016) demonstrated that Random Forest performed better than neural networks when temporal dynamics were not volatile and feature engineering was rigorous which is the case for current study when using 168-hour lagged values, rolling statistics and Fourier terms. Hahn et al. (2009) also showed that the non-linearity of multiple weather variables is better modeled by ensemble methods than by parametric models which is consistent with the primary relationship found in the Finnish data between temperature and load.

The poor performance of LSTM compared to Marino et al. (2016) and Kong et al. (2017) is surprising. These authors found LSTM to outperform ANN and SVR models on residential and building loads. The main difference between the results here and the residential studies is likely the size of the datasets. In the residential studies, datasets were derived

from hundreds of homes over several years resulting in tens or hundreds of thousands of training sequences while in this study the dataset is a single national load series with data spanning around 1 year. Jain & Gupta (2024) explicitly found that deep learning models performed poorly compared to simpler baselines when the amount of training data was too small and hyperparameters were not tuned which is true in this case.

The CNN training loss history suggests overfitting meaning that the loss to the training set approaches zero after 2 epochs while the loss to the test set increases and does not recover. This demonstrates that the convolutional filters fit the intra-day variation from the training split without transferring to the test set. This observation is on point with the warning of Lago et al. (2018) that the deep learning hybrids only show stronger performance than traditional statistical methods with strong feature engineering and rigorous temporal cross-validation of which only one is sufficient to avoid overfitting with small sized data.

The low performance of SVR against expectations of the literature benchmarked by Chen et al. (2004) and Pai & Hong (2005) is due to the training set restriction. The kernel matrix computation of the full training set was intractable with the available computing resources, so the SVR was trained on random subsets of the training set of 5,000 samples. This probably led the SVR to underestimate the long-term seasonal trend of the total training period including the winter peaks in the load leading to a higher RMSE than to be obtained by training model on all the data.

The feature correlation analysis found that the Finnish electricity is primarily driven by temperature with a Pearson correlation of -0.85. This confirms with the systematic evidence found by Hong & Fan (2016) that temperature is the most important meteorological variable in nearly all forecast horizons and locations. The virtually perfect correlation between wind speed and wind gust suggests that the latter variable can be removed or replaced by a new variable that combines both to eliminate information redundancy in the feature matrix while retaining useful information.

5.5 Methodological Limitations

5.5.1 Size and Time Period of the Dataset

The main methodological challenge of this work is a short time window of around two (2025-2026) years of data. This timeframe spans two seasonal cycles but is short to capture inter-annual variation in the Finnish electricity demand due to macroeconomic-wise change in industry structure and electrification of residential heating and transport sector. Such generative deep learning algorithms require much larger training sequence which is typically five or more years of hourly data to converge with stable multi-year representations of the seasons.

5.5.2 Single Weather Station

This thesis used weather data from just a single station Helsinki-Malmi which may underrepresent the weather dynamics over Finland's geographically wide distribution. Finnish electricity consumption is a composite of electricity use from regions that are over 1,000 kilometers apart in latitude and which experience significantly different temperature and daylight conditions. An aggregated weather variable source from a selection of FMI stations across different Finnish climate zones would be better representation of load-weather relationship across Finland and minimize the unexplained errors due to the geographical average.

5.5.3 No Hyperparameter Optimization of Deep Learning

Deep learning models used a variety of architectures found in the literature, but no hyperparameter optimization was conducted by grid search or Bayesian approaches. Particularly the LSTM model could significantly benefit from increased training epochs or the use of sequence lengths of other than 24-timestep window, bidirectional recurrence or attention mechanism. The fact that the CNN exhibited overfitting very quick and the LSTM appeared to converge very poorly suggests that the models were not operating in the hyperparameter sweet spot.

5.5.4 Point Forecasting Only

This study only generated point forecasts and used point accuracy metrics like RMSE, MAE, MAPE and SMAPE to evaluate the forecasts. As pointed out by Hong & Fan (2016), point forecasts may no longer be sufficient in today's power grids where renewable intermittency creates significant uncertainty. This study's result therefore cannot directly support reserve capacity planning or risk-weighted dispatch which requires probability forecasts.

5.5.5 SVR Computational Restriction

The SVR was only trained on 5,000 randomly selected samples from the training data set because of the kernel matrix which represents the training data and has a size equal to the number of training samples. It has a computational complexity of O(n2) where n is the number of training samples. This places a selection bias on the model to learn the global seasonal patterns of load. Therefore, the performance of SVR should only be considered as a lower bound estimate of what can be achieved with this learning system rather than the answer to whether the RBF kernel should be used for load forecasting.

5.6 Future Work

The limitation in this study suggests many different avenues for future research which would contribute to this thesis.

First, expanding the training data to include at least five years of hourly data from Fingrid with multi-station FMI weather data would make the data environment for the LSTM and CNN-LSTM to perform at their full potential. A follow-up study on this data set would answer the question of whether the deep learning can outperform Random Forest on Finnish national load data.

Second, extending the forecasting framework to provide probabilistic forecasts would enable well-calibrated error bars. This addition is directly applicable to Fingrid's business

needs to control reserve capacity under uncertainty due to increasing amounts of solar and wind generation in Nordic electricity system.

Third, a richer feature engineering process could be developed to include the demand side. These predictors include hourly electricity spot market prices from the Nord Pool market and electric vehicle charging load data from the Finnish Transport and Communication Agency (Traficom) and distributed solar generation from the Finnish Energy Authority (FEA). These would capture the non-linear multi-scale interactivity in demand identified in the literature review to pose challenges for traditional statistical models.

Last, the transfer learning could be used to transfer knowledge from a deep learning model trained using a large European series such as the ENTSO-E Transparency Platform hourly load data from all 36 member countries and fine-tune the model for the Finnish Fingrid series. This would enable deep learning models to benefit by data from similar Nordic load profiles.

6 Conclusion

This thesis attempts to provide a controlled evaluation of six machine learning models for electricity load forecasting. The process is done using data from Finland's national grid with the three main goals that distinguish this work from previous machine learning literature in this field. Those three goals include the process to control for cofounding factors when comparing machine learning models, to evaluate all model's performance across multiple time horizons and to disentangle the effects of model's architecture choices from alternative representations of features.

Different documents containing an identical preprocessed dataset of hourly energy load from the Fingrid and Finnish Meteorological Institute (FMI)weather variables from 2025-2026 were used to train six models: Linear Regression, Support Vector Regression, Random Forest, LSTM, CNN and CNN-LSTM. The models share both feature engineering and data splitting as well as identical four evaluation metrics. They were then evaluated on a stratified train-validation-split in chronological order, so that any observed differences are due to architectural features of the models not the experimental effects.

The key contribution of this paper is that Random Forest provided the most accurate forecast across all three forecasting horizons and all four measures of prediction performance. Random Forest with overall RMSE of 343.41 MWh and MAPE of 2.58% represent improvement over the Linear Regression of about 59% and 75% respectively and over the best performing deep learning models (CNN) of about 66% and 78% in RMSE and MAPE respectively. Random Forest's superior performance is due to its non-parametric adaptability to the highly non-linear temperature-load relationship, its resistance to collinearity through random feature selection and efficient re-training when new operational records become available.

CNN being the best performing deep learning neural network with RMSE of 1,012.89 MWh and MAPE of 11.69% was followed by CNN-LSTM hybrid with RMSE of 1,357.77 MWh and the LSTM with RMSE of 2,161.09 MWh. The inability of the LSTM to

outperform even the Linear Regression baseline is explained by a training instability that is caused by the small training set that may not provide enough training sequences for the recurrent layers to learn a stable gradient flow during training with an input sequence of 24 timesteps. This result is on point with the literature showing that deep learning models are expected to use much larger data sets than in this study to achieve their full potential.

The partitioned analysis by horizon showed that model performance is quite consistent across 24, 168, 720-hour window with Random Forest consistently being ranked number one. CNN and Linear Regression showed lower RMSE at the 720-hour horizon compared to 24 hours due to low variability trend-dominated long-term signal opposite to the high variability intraday signals of the short-term window. This finding has system operational implications for grid operators who need to choose forecasting tools for various operational planning tasks.

The main limitations of this study show that there are different parts of load forecasting to be studied in more depth. The time frame used in this study is considered too short for deep learning models to learn year-to-year patterns. The site-specific observations of weather from the Helsinki-Malmi station may not capture Finland's highly variable weather. Since hyperparameter optimization has not been done for deep learning models, the underperformance of LSTM cannot be explained as unsuited architecture. Finally, the focus on point forecasts and related metrics has left out the uncertainty quantification that is increasingly required for renewable-integrated electricity system.

Overall, the findings of this thesis offer a practical advice to researchers on machine learning model selection for electricity load forecasting in Finland. Random Forest, with a dense feature engineering pipeline, is preferred architecture for operational applications with limited computational resources and training period of 12-24 months of hourly input data. For researchers with access to over five years of hourly load and weather observations across multiple weather stations capable of hyperparameter

searches, CNN and Transformer-based models should be prioritized. The experimental design proposed in this thesis with along with the Fingrid-FMI data pipeline, feature engineering taxonomy and the horizon-stratified evaluation protocol offers a benchmark to test model innovations in electricity load forecasting for Finland.

References

Almeshaiei, E., & Soltan, H. (2011). A methodology for Electric Power Load Forecasting. *Alexandria Engineering Journal*, 137-144.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32. https://doi.org/https://doi.org/10.1023/A:1010950718922

Chen, B.-J., Chang, M.-W., & Lin, C.-J. (2004). Load forecasting using support vector machines: A study on EUNITE Competition 2001. *IEEE Transactions on Power Systems, 19*(4), 1821-1830. https://doi.org/https://doi.org/10.1109/TPWRS.2004.835679

Dudek, G. (2016). Pattern-based local linear regression models for short-term load forecasting. *Electric Power Systems Research, 130*, 139-147. https://doi.org/https://doi.org/10.1016/j.epsr.2015.09.001

Fan, S., & Hyndman, R. J. (2012). Short-Term Load Forecasting Based on a semi-Parametic Additive Model. *IEEE Transactions on Power Systems*, 134-141.

Hahn, H., Meyer-Nieberg, S., & Pickl, S. (2009). Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research, 199*(3), 902-907.

Harvey, A. C. (1991). Forecasting, Structural Time Series Models and the Kalman Filter. *The Journal of the Operational Research Society*.

Hippert, H. s., Pedreira, C. E., & Souza, R. C. (2001). Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. *IEEE Xplore, 16*(1), 44-55. https://doi.org/https://doi.org/10.1109/59.910780

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *PubMed, 9*(8), 1735-1780. https://doi.org/https://doi.org/10.1162/neco.1997.9.8.1735?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

Hong, T., & Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting 32(3)*, 914-938.

Jain, A., & Gupta, S. C. (2024). Evaluation of electrical load demand forecasting using various machine learning algorithm. *Frontiers in Energy Research, 10*.

https://doi.org/https://doi.org/10.3389/fenrg.2024.1408119?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

Kong , W., Dong, Z. Y., Hill, D. J., Jia, Y., Yan, X., & Zhang, Y. (2017). Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid, 10*(1), 841-851. https://doi.org/https://doi.org/10.1109/TSG.2017.2753802

Lago, J., Ridder, F. D., & Schutter, B. D. (2018). Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy, 221*(4), 386-405. https://doi.org/https://doi.org/10.1016/j.apenergy.2018.02.069

Lecun, Y., Bengio, Y., Haffner, P., Rachmad, Y. E., & Bottou, L. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedingd of the IEEE, 86*(11), 2278-2324. https://doi.org/https://doi.org/10.1109/5.726791

Li, C., Ding, Z., Zhao, D., Yi, J., & Zhang, G. (2017). Building Energy Consumption Prediction: An Extreme Deep Learning Approach. *Energies, 10*(10), 1525. https://doi.org/https://doi.org/10.3390/en10101525?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

Marino, D., Amarasinghe, K., & Manic, M. (2016). Building Energy Load Forecasting using Deep Neural Networks. *IEEE Industrial Electronics Society.* https://doi.org/https://doi.org/10.1109/IECON.2016.7793413

Pai, P.-F., & Hong, W.-C. (2005). Forecasting Regional Electricity Load Based on Recurrent Support Vector Machines with Genetic Algorithms. *Electric Power Systems Research, 74*(3), 417-125. https://doi.org/https://doi.org/10.1016/j.epsr.2005.01.006

Park, D. C., El-Sharkawi, M. A., Marks II, R. J., Atlas, L. E., & Damborg, M. J. (1991). Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems 6(2)*, 442-449.

Taşcıkaraoğlu, A., & Uzunoglu, M. (2014). A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable*

Energy Reviews, 34(3), 243-254. https://doi.org/https://doi.org/10.1016/j.rser.2014.03.033

Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*.

Vianita, E., & Tantyoko, H. (2025). A Comparative Study of Machine Learning Models for Short-Term Load Forecasting. *JURNAL MASYARAKAT INFORMATIKA, 16*(1), 93-103. https://doi.org/https://doi.org/10.14710/jmasif.16.1.73130

Wang, Y., Gan, D., Sun, M., Zhang, N., Kang, C., & Zongxiang, l. (2019). Probabilistic Individual Load Forecasting Using Pinball Loss Guided LSTM. *Applied Energy, 235*, 10-20. https://doi.org/https://doi.org/10.1016/j.apenergy.2018.10.078

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation, 31*(7), 1235-1270. https://doi.org/https://doi.org/10.1162/neco_a_01199?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

Zhang, P. G., Patuwo, E., & Hu, M. Y. (1998). Forecasting With Artificial Neural Networks: The State of the Art. *International Journal of Forecasting , 14*(1), 35-62. https://doi.org/https://doi.org/10.1016/S0169-2070(97)00044-7

Appendix 1- Full Source Code

Full code can be found at: https://colab.research.google.com/drive/1jLxMmSVD_-2IAgeQgrGjJU0dL1XYMln6#scrollTo=Fz0QwitYH2W2