TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Mohammad Arif-Uzzaman (194365IASM)

"FORECAST OF THE ELECTRICITY CONSUMPTION DURING PANDEMIC TIME"

Master's thesis

Supervisor: Dr. Juri Belikov
Department of Software Science

Co-supervisor: Abiodun Emmanuel Onile

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

M Mohammad Arif-Uzzaman (194365IASM)

"ELEKTRITARBIMISE PROGNOOS PANDEMEMIA AJAL "

Magistritöö

Juhendaja: Dr. Juri Belikov
Tarkvarateaduse osakond

Kaasjuhendaja: Abiodun Emmanuel Onile

Tallinn 2021

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Mohammad Arif-Uzzaman

10.05.2021

Abstract

There has been a major shift in electrical load prediction during the COVID-19 pandemic. For this reason, there is a big change in the electricity grid operation because, during peak or off-peak hours, the consumption levels change very unexpectedly and quickly. In another geographic way, our electricity consumption behaviour changes or varies by region during a pandemic. There is a need to develop an adaptive/robust prediction model to maintain an adequate load forecast for future demand. Such as LSTM and XGboost, to reduce prediction error.

This thesis presents the concept of the LSTM (Long Short Term Memory) model and the XGBoost (Extreme Gradient Boosting) model, which allows daily, weekly, weekend, and monthly use according to the actual total load from 1 December 2019 to 31 May 2020. The XGboost and LSTM models used the only function as the actual load input in the three countries. The entire six months of data were performed before and during the lockdown in the forecast for hours, weeks, weekends, and months in the LSTM and Ensemble XGBoost. In contrast, the RMSE, STD, loss function, and MAPE were calculated from this continuous prediction and the actual total load error in the test data.

In summary, the actual total exposure of the Estonian, Finnish and Norwegian error prognoses was observed in various parameters. However, there are various errors for future predictions before and during the lockdown time. The final LSTM model achieved the lowest RMSE, STD, Loss, and MAPE values up to respectively. The ensemble XGboost is the predictive model that took the longest to process. In contrast, the XGBoost model with the second-lowest RMSE, STD, Loss, and MAPE values took the least time to set the parameters and calculate the predictions during a pandemic.

This thesis is written in English and is 85 pages long, including 5 chapters, 38 figures, and 12 tables

List of abbreviations and terms

LSTM	Long Short-term Memory
XGBoost	eXtreme Gradient Boosting
COVID	Coronavirus Disease
RMSE	Root Mean Square Error
STD	Standard Deviation
MAPE	Mean Absolute Percentage Error
EMS	Energy Management Systems
FF-ANN	Feedforward Artificial Neural Network
MLP-ANN	Multilayer Perceptron Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
SARS-CoV-2	Severe Acute Respiratory Syndrome Coronavirus 2
MLR	Multiple Linear Regression
ML	Machine Learning
ANN	Artificial Neural Network
GA	Genetic Algorithm
RF	Random Forest
AI	Artificial Intelligence
API	Application Programming Interface
ENTSO-E	European Network of Transmission System Operators for Electricity

List of figures

List of tables

1 Introduction

In the present time, we can observe a difference is coming for this COVID-19 pandemic time in Electrical energy load forecasting. In this pandemic time, people stay at home, and they are not using a basic amount of Electricity. For this reason, a big change comes in power grid operators because on peak hour or off-peak hour, the level change very unexpectedly. On the contrary, people are maintaining social distance, and for this reason, power consumption is shifting very faster. In a different geographical way, our electricity consumption is changing every day in pandemic time. In a short time, a big change is making difficulties for Electric load forecasting. We will work with a new system that calculates economic activities, forecasting, and manages future load patterns. In this paper, we need to develop a new method to maintain a reasonable load forecast for future demand. Then, there are no unexpected situations for the local electricity grid. Without expecting the maintenance burden, local energy producers will make mistakes, which will damage our energy system and the state's financial damage. The electricity grid cannot become a promising technology that facilitates the future electricity grid and balances supply and load demand.

During the pandemic, consumer habits are changing over time. Still, the COVID-19 epidemic forces consumers to change their lives, practices, and shopping habits worldwide at an unprecedented rate and scale. Electricity demand fell to weekly levels due to the freezing, and an increase in residential use only partially offset the sharp decline in services and industry.

In this current economic period, a load forecast is needed for the energy sector, as it does not include a load forecast. Many different mathematical methods have been developed to predict the load, but in our research, we try to find a simple process in the long run.

1.1 Background Information

During the COVID-19 pandemic, there is tangible evidence of low electricity demand. At this point, all smaller electricity networks are facing predictable errors and internal fluctuations in net energy. Reducing electricity consumption is very important in this pandemic time. In particular, we introduce changes in consumption in Estonia, Finland, and Norway. Here we are trying to find out the differences in consumption explained by pandemic time uses. We also figure out the actual voltage and frequency fluctuations. One of our main goals is to reduce the energy consumption expected as a pandemic develops, and this will have a big impact on the functioning of small electricity networks.

Another problem is that at the moment of the pandemic, the reduction in energy consumption of small electricity networks is large, which will cause problems in the future. It is not clear how a global pandemic could affect electricity networks in the future, as it is believed that new methods or techniques could be used to address potential future pressures. One of the key findings is that reducing energy consumption during a pandemic is critical and has a significant impact on network performance. Low power consumption usually affects the operation and control of generators and can lead to voltage and frequency deviations and, in general, reduced reliability and flexibility. In this thesis paper, we are trying to understand our new method or algorithm for load prediction.

1.2 Statement of the Problem

Due to this pandemic, time load forecasting impacts purchasing, generating, and electricity distribution. One of the difficulties to maintenance local energy supplier bugs and fix financial damage in the energy sector. In contrast, measurement of economic activities of future load forecasting in any pandemic time. Difficulties in obtaining accurate data on time behaviour in the event of a pandemic due to changes in factors such as prices and the corresponding demand due to price changes. A utility may suffer if it does not understand the error level in estimating short-term exposure during a pandemic and makes decisions. There is a difference in behaviour between consumers using different meters, especially smart and conventional meters, and between separate tariffs. The utility must understand this and develop its forecasting model for each metering system and then add it to the final value of the forecast. Otherwise, you will receive an inaccurate prediction.

1.3 Scope of the project

They are analysing humanity and mobility data for improving load forecasting. Develop a forecasting algorithm for reliable load forecasting in any long period of pandemic time. They are making a system that will help maintain future power grids by ensuring supply and demand stability. Develop a XGBoost model that can help to plan for the future expansion (in terms of size, location, and type) of the power system. When identifying areas or areas of high or increasing demand, utilities are likely to generate electricity close to the load. Lead to fixing the uncertainty situation in load forecasting in pandemic time. Load forecasting ensemble methods will price elastic consumption behaviour will be analysed.

1.4 Aim and Objective of the project

Before this pandemic time, our electrical energy system was balanced, but the energy consumed system change suddenly this time. For this reason, energy use was low in pandemic time, and it puts big pressure on our local power grid. Energy load forecasting also makes a financial loss for a country because they produce the same amount of energy. Still, they cannot use their full amount of energy, and they are losing their natural resource proper use. Long-term forecasts play a key role in policymaking and capacity building. As the new technologies and policies affect demand itself, combined methods are usually used to include as many relevant factors as possible. These factors include consumer behaviour, the impact of technology adoption, and simulated scenarios. On the contrary, which countries are lifting electrical energy then they stored their full of their storage system then lose their extra energy. At that time, energy prices decreased, and it makes a bad financial impact on every country.

This study aims to explore a new algorithm or method for energy load forecasting problems. The load forecasting method or model is necessary always will have the best results for the energy management system. The new algorithm for energy load forecasting will always perform better in any pandemic time. The present research work identifies some variations of load forecasting for a long period: i) Humanity and mobility, ii) long-time algorithm, iii) Supplier bugs from the small power grid, and iv) Measurement of future load forecasting. All forecasting problems are explored for pandemic time or COVID-19. For new methods, only a long period of forecasting problems has been learned. The electrical energy load forecasting using evaluation metrics: root mean squared error (RMSE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (SMAPE). These metrics help to explore the errors from different aspects. While deciding the long period of energy load forecasting, all these evaluation metrics should be considered overall in any pandemic time.

This thesis considers a new method or algorithm a long period of energy load forecasting techniques for energy forecasting in any pandemic time. In this project, we discuss the error in the forecasting model and make a new algorithm for forecasting and forecasting problems. We also give a general understanding of forecasting models in RMSE, explain the difference between normal time and lockdown time of load forecasting.

1.5 Motivation and Research Question

Global population growth and energy availability are currently the main reasons for worrying about electricity consumption. Various simulation tools, algorithms, and estimation matrices have been used to predict the optimal demand for electricity. While previous load forecasting methods use dynamic equations to indicate, ensemble methods use historical data to predict future direction. However, the modelling of power demand models for robust solutions has not yet been developed, as the existing techniques are only useful for solving long-term dependencies. Additionally, secret methods are static as they are based on purely historical data. This project proposes a comprehensive learning framework for predicting electricity demand, taking into account long-term historical dependencies. First, a cluster analysis of all monthly electricity consumption data is performed to obtain seasonally segmented data. A description of the load forecast is then made to get a more detailed picture of the power lost during the pandemic.

From the above overview there are three groups of research questions this work will be seeking answers for:

1. Find a suitable method for managing the electrical load to predict a pandemic load:

· How to the measurement of economic activities of future load forecasting in pandemic time?

· What types of energy load forecasting model is optimal for various periods (for long-short period) of any pandemic time?

· How to manage load forecasting impacts on purchasing, generating, and energy distribution in pandemic time?

· How to maintenance local energy supply operation parameter during pandemic and fix financial damage?

2. Adjust the existing system:

· What is the other system of possible improvements that can be made to improve the new method in a long-short time in a pandemic?

3. Further research:

· Are there other ways to improve energy load forecasting for a long-short period of pandemic time?

1.6 Expand Literatures

In all, by this research paper, we can understand the energy market in pandemic time and how small power grid affected their system. It also makes global crises. In this paper, they are focus on pattern, consumption, and stability. This research paper can understand how the Estonian power grid was supplying pattern change in COVID-19. In pandemic time, change the voltage and frequency and try to figure out the normal time and pandemic time pattern. One of the most important things is that reducing energy use is vital during a pandemic and has a huge impact on the operation of small power grids. Another problem is that we have reached record levels due to the relative pandemic share of renewables. This event can help us better understand the impact of a high percentage of renewable energies on small grids and provide insight into a prosperous future—renewable energies.[1]

Electrical energy consumption and the energy market in Poland during the COVID-19 pandemic This literature expand on how the COVID-19 pandemic has affected the Polish energy market. This included an analysis of electricity demand and consumption during the recession. All data were compared with previous characteristic periods. The article also contains an analysis of price changes on the Polish power exchange. The COVID-19 pandemic and the economic downturn have affected Poland's electricity consumption.[2] The electricity demand peak was 6.4% lower than in 2019 and 10% lower than in 2020. The average rise in electricity demand was 7.8% lower than in 2019 and 2018. The decrease in electricity demand may be caused by other factors, such as relatively warm winters and early spring. However, the analysis of energy consumption shows a significant drop in need, especially in the second half of March, when most of the restrictions came into force.[24]

Optimal Day-Ahead scheduling and operation of the prosumer by considering corrective actions based on very Short-Term load forecasting in Energy management systems (EMS) play an important role in the optimal functioning of consumers.[3] This article recommends a new daily routine and schedule for consumers using a two-step corrective LF. At the first level, LF, MLP-ANN time series are predicted based on the historical values of the proposed method. Predicted data, not predefined values, optimize consumer performance and time. Besides, real-time measurement is performed. If the expected load data inaccuracy concerning the measured data exceeds the desired threshold value, the second stage LF is started. The FF-ANN is used to adjust the LF using the load samples from the last 30 days as input and the actual load data from the working day as the target.

LSTM architectures for energy Time-Series forecasting is a very active area of research, as reliable information on future electricity generation enables the safe operation of the electricity grid and helps reduce over-production. As redundant neural networks go beyond most machine learning predictions for predicting time series, they have become widely used models for energy prediction problems. In this article, the permanent measurement prediction and the ARIMA model as basic methods and short-term memory (LSTM) based neural networks with different configurations for implementing a multi-level energy prediction are created.[4]

Grouping the population according to different periods of fluctuating loads and aggregate forecasts can increase the accuracy of load forecasts. However, the strength of the load fluctuations reflects the other behaviour of the electricity consumption of the population and influences the results of the population cluster. Statistical experiments determined the optimal results for summing the full load for the different fluctuation periods. Finally, a random forest forecaster is selected based on the ensemble's learning. Based on the optimal aggregation results of the different fluctuation periods, a mobile forecasting model was created, which presents a forecast of yesterday's total load in the apartment distribution network.[25] Implications of COVID-19 for the electricity industry: A comprehensive review allocates COVID-19 worldwide; human activities have changed significantly. In this situation, it will seriously affect the electricity sector and face major challenges. This document provides a comprehensive overview of the impact of the pandemic on the electricity sector. Electricity demand has fallen dramatically as governments worldwide have implemented lock-in restrictions, while load composition and daily load profile have changed. The electricity market is also badly affected, while long-term investment in clean energy should be stable. Externalities, such as emission reductions, are also discussed.[26]

Monitoring of electrical consumption, including self-isolation during the COVID-19 pandemic, also shows the most important task is a reasonable calculation of the electrical capacity of residential buildings and public buildings. Roselectromontazh's studies showed a significant difference between the actual and calculated electricity capacity, which is further confirmed by the reports of companies from the electricity networks. This allows a considerable reduction in the difference between the actual and calculated electrical power. From 2020, large construction companies in the Republic of Tatarstan will apply updated values to reduce the cost of building utilities. For power companies, the savings reduce electricity losses and "blocked electricity" and eliminate inefficient investments. It is important to note that the specific values of the electrical load are calculated, taking into account the summer and winter peaks to avoid emergencies. However, it was impossible to predict when people would need to be isolated at home to prevent the spread of SARS-CoV-2 (COVID-2019), a coronavirus associated with the severe acute respiratory syndrome. To determine the impact of electricity uses in residential areas during the quarter, appropriate regulations for 2019 have been developed.[5]

Impact of COVID-19 and nine-minute call on Indian power sector of blackout prevention research on the effects of the initial phase of COVID-19 on the Indian energy sector. This study will help researchers working in this field who want to mitigate the long-term impact of COVID-19 on the Indian energy sector. Additional studies should also be performed shortly to reduce the risks if more data are available. The Indian energy sector is also not immune to this pandemic. However, as negative effects are already being seen in the energy sector, the nine-minute speech drew more media attention. This article discusses the impact of this pandemic on India's energy sector in mitigating its long-term effects. This study may help prepare for such a situation in the future.[27]

1.7 Consumption Behaviour

Pandemics caused by COVID-19 are widespread in most countries around the world. They consider that the energy sector is one of the fundamental parts of modern society, and it’s significantly affected. During the pandemic, this was done through research and analysis of various aspects. Users’ habits tend to change over time, but the COVID-19 epidemic forces consumers to change their lives, patterns, and shopping habits at a step and scale never before seen in the world. Electricity demand dropped to Sunday levels under lockdown, with dramatic reductions in services and industry only partially offset by a higher residential use.[6]

Estonia, Finland and Norway has daily demand for energy, and in this pandemic time, the local power grid cannot manage its maintenance. This is because of a low level of energy consumption. So, for this reason, we are losing our energy and our natural resources. Another main energy is our oil energy, and we also messed up our energy consumption system.

In 2019 was the use of energy was bigger than in 2020 at the same period. In this COVID-19 time, people don’t use their cars, and for this reason, electricity companies lose lots of natural resources because they store their energy. When they stop their energy lifting, then that energy capacity will go down. That’s why for this reason, they need new systems.

2 Methodology

Short-term load forecasting is playing a key role in the implementation of smart power grids. They are needed during a pandemic to optimize the various possible network management solutions, including integrating low power consumption and the future use of managed energy, despite the need for precision. In estimating energy load, much of the project has focused on individual energy consumption or on sets of such data.

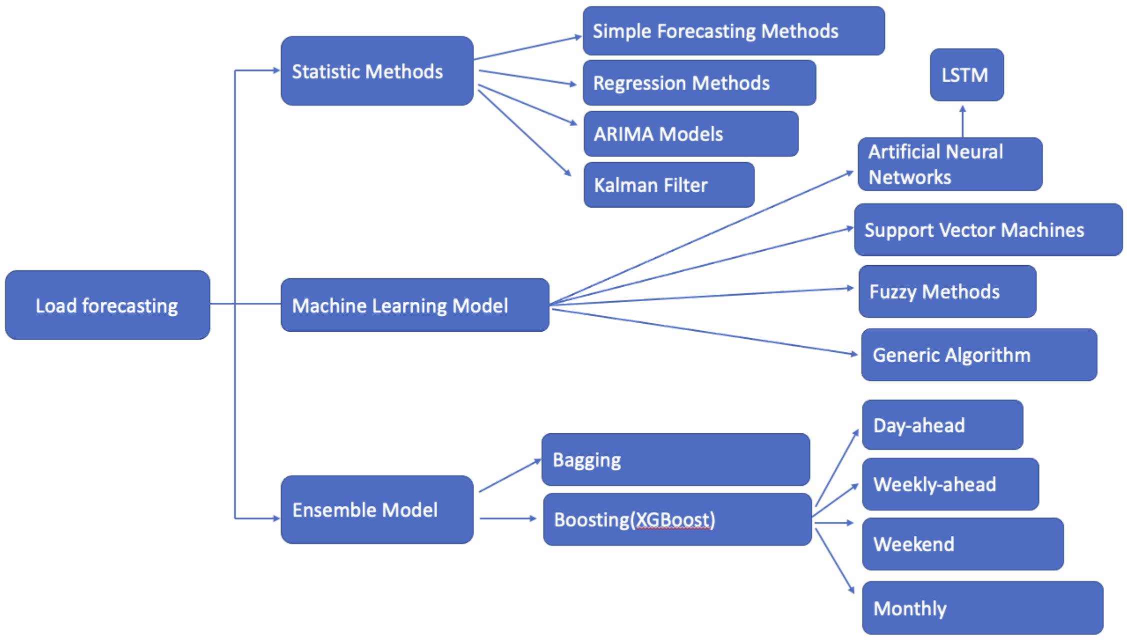

Figure 1. Architecture of time series predicting model

2.1 Statistical Methods

Statistical methods are used to identify the explained variables and to predict future loads (e.g., regression models) (See figure 1). Time-series methods use past data for future estimates. In other words, the past model of the predictor is predicted for the future because what will happened in the past is likely to happen in the future. Sometimes time series models are extended to include predictor variables.[7]

2.1.1 Simple Forecasting Methods

There are simple prediction methods that can be used to solve simple prediction problems or, with more complex prediction methods, to take special cases into account. Average method - the forecast is the average value of historical data from time series, which can be calculated as follows:

$$y^\wedge{}_{T+h|T} = y = \underset{T}{(}y^1 + \cdots + y^T)/T \quad (1)$$

where $\hat{y}_{+h|T}$ is the h-step forecast of y_{T+h}, taking into account all observations $y_1,...,y_T$ up to time T. [8]

2.1.2 Regression Methods

In summary, the above-mentioned polynomial and non-additive regression models are special cases of the MLR. Therefore, MLR can be used to generate a large number of non-linear response surfaces. In other words, linear models are considered linear in their parameters and can include large variations in the relationship of variables through transformations applied to explanatory variables.

Regression is one of the most useable statistical techniques. Regression analysis examines the relationship between two or more variables (predicted and planned). Regression models predict values considering future values of past (backlog) or explanatory variables of predicted variables. The principle of regression models is a linear relationship between the forecast and the explained variable. Regression equation can be written as follows:[9]

$$y_t = f(x_t) + \varepsilon_t, \quad (2)$$

where yt is forecast variable, $f(x_t)$ linear regression function, εt random error component. The term "error" does not mean an error but a deviation from the basic linear model. Save anything that may affect y_t except x_t. The ratio can be positive or negative and indicates whether the value of y moves up or down the axis as x increases by one.

An in-depth study was performed using a regression-based approach. The most common regression methods are:

- Simple linear regression measures the relationship between a predicted variable and an explained variable and can be defined as follows:

$$y_t = \beta_0 + \beta_1 x_1 + \varepsilon_t, \ (3)$$

where y_t is forecasted variable, x_1 explanatory variable, β0 is x intercept (pre- dictated value of y when x = 0), β_1 is the average predicted change in y resulting in a one-unit increase in x and εt is a random error component.

- Multiple linear regression measures the relationship between predicted and multiplied explained variables. The MLR equation can be written as follows:

$$y_t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon_t, (4)$$

where y_t is the forecasted variable, x_1, . . . , x_k are the k explanatory variables, β0 is x intercept (predicted value of y when x = 0), β_1, . . . , β_k measure marginal effects of the explanatory variables and εt is a random error component. [10]

- Polynomial regression addresses the problem of nonlinear relationships of variables. In this case, the data are regressed on the polynomials to match the polynomial equation. In other words, polynomial regression models include polynomial explanatory variables that make the response function curvilinear. For example, if the load yt is predicted by a regression model of a polynomial with the declared variable xt and the sequence of this polynomial is k, the following model may be considered:

$$y_t = \beta_0 + \beta_1 x_t + \beta_2 {x^2}_t + \cdots + \beta_k {x^k}_t + \varepsilon_t. \ (5)$$

- A non-additive regression model is used when the effect of an explanatory variable depends on the level of another explanatory variable. In this case, the interaction effects are included in the regression model by multiplying at least two explanatory variables. An example of a non-additive regression model with two explanatory variables x1, t and x2, t is:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{1,t} x_{2,t} + \varepsilon_t. \ (6)$$

In summary, the polynomial as mentioned above and non-additive regression models are special cases of the MLR. Therefore, MLR can be used to generate a large number of non-linear response surfaces. In other words, linear models are considered linear in their parameters and can include large variations in the relationship of variables through transformations applied to explanatory variables.

2.1.3 ARIMA Models

Box and Jenkins developed a mathematical model for predicting time series by adapting them to the data and using a model suitable for prediction (i.e., the ARIMA model). In general, the ARIMA process is written ARIMA (p,d,q), where p is the number of autoregressive sequences in the model. Autoregressive orders provide previous values in a series that are used to predict current values. Difference (d) indicates the charge of differentiation applied to the series before model evaluation. The moving average (q) shows how deviations of previous values from the series average predict current values. Therefore, the model is commonly referred to as the ARMA (p, q) model, where p is the order of the autoregressive part and q is the moving average.

$$Xt = c + \phi 1Xt - 1 + \phi 2Xt - 2 + \cdots + \phi pXt - p + \theta 1\varepsilon t - 1 + \cdots + \theta q\varepsilon t - q + \varepsilon t (7)$$

The ARIMA model (p, d, q) is a generalization of the ARMA model, where p, d, and q are non-negative integer values relative to a sequence of autoregressive, integrated, and movable parts of the model.[11]

2.1.4 Kalman Filter

By achieving load prediction accuracy, the load prediction model uses state space and Kalman filtering technologies to reduce the difference between actual loads and predictions (random error). This approach introduces the periodic charge component as a random process. Data older than 3 to 10 years are needed to calculate the frequent change in load and to estimate the electrical system-dependent variables (load or temperature). The disadvantage of these methods is that it is difficult to avoid observational noise in the prediction process, especially with multiple variables. The evaluation of the Kalman filter parameter was performed.[12]

2.2 Machine learning Methods

AI is the technology of training machines to perform human tasks for future. The origins of artificial intelligence can be traced back to the 1950s when scientists looked for ways to enable computers to solve problems themselves. In other words, artificial intelligence is when machines are given similar human characteristics.

Machine learning (ML) is a specifically subset of artificial intelligence that teaches machine learning. ML models look for patterns in the data and try to conclude. Using this example, ML can learn how to use this example. When the algorithm gets the right result, it applies this knowledge to new data sets.[13] The ML life cycle can be represented as follows:

- Asking a question.
- Collecting data.
- Practicing an algorithm.
- Gathering feedback.

Using feedback to improve an algorithm.

Artificial intelligence and ML require very large and very different amounts of data to find and learn patterns. The ML type is:

Guided learning refers to a set of data with a response variable (also called a label). The answer can be continuous or categorical. The algorithm learns the response variable based on the predicted predictor variables.

Unguided learning means that the database does not have a response variable. There is no help in learning predictor variables. Knowledge is based on the similarities or distances between each row in a data set.

Semiconductor learning deals with a set of data with partially filled response variables. These methods are based on the idea that, although members of an unmarked data group are unknown, this data contains important information about group parameters.

Gain is an ideal choice in cases where only the initial state of the data is available as input, and there is no single answer but many possible outcomes. Supervised and unsupervised learning algorithms require clean and accurate data to achieve the best results.[13]

2.2.1 Artificial Neural Networks

ANN consists of parts of a process called neurons. An artificial neuron has more than one input and one output that are interconnected. Network processing capability is an advantage of stored node connectivity, called scales. These weights are obtained by learning or adapting through several training models. Neurons are divided into three layers:

- The input layer contains training data.
- The hidden layer implements the activation function.
- The output layer responds.

Depending on the complexity of the ANN, there is an input and output layer and several hidden layers. The levels use the trigger functions to change the received data before taking it to the next level. Activation functions allow the ANN to model complex nonlinear relationships between parts. The main limitation of ANN is over-regulation (high-precision training), in which case the model may not be suitable for other predictions. ANN training takes time. There are two ways to learn ANN:

- Front doors: the input data is distributed layer by layer towards the last layer that outputs the forecast,
- Rear doors: each net weight is adjusted in proportion to its effect on the total error.[14]

2.2.1.1 LSTM

The Long short-term memory(LSTM) is an artificial recurrent neural network for deep learning. LSTMs are very powerful in sequence prediction problems because they can store past information. LSTM is important in our case because the previous value of a property is crucial in predicting its future demand. The goal of the sequence prediction project is to arrive at a final destination value where that value is defining by data before lockdown and during the lockdown.[21]

$$it = \sigma(\omega i[ht - 1, xt] + bi)$$
$$ft = \sigma(\omega f[ht - 1, xt] + bf)$$
$$ot = \sigma(\omega o[ht - 1, xt] + bo) \quad (8)$$

Here, $it \rightarrow$ represents input gate.$ft \rightarrow$ represents forget gate.$ot \rightarrow$ represents output gate.σ $\rightarrow$ represents sigmoid function. $\omega x \rightarrow$ weight for the respective gate(x) neurons. ht-1 $\rightarrow$ output of the previous lstm block(at timestamp t-1).$xt \rightarrow$ input at current timestamp.b$x \rightarrow$ biases for the respective gates(x).[28]

2.2.2 Support Vector Machines

In Support Vector Machines (SVM) looks for the optimal hyper-plane separating the two classes. A hyperplane is a line in a p-dimensional space in two dimensions, and a hyperplane is a flat affine subspace of hyperplane dimension p–1. It finds the optimal hyperplane by maximizing the margin between the closest points of the two classes. In two-dimensional space, the points lying on the margins are called support vectors, and the line passing through the midpoint of the margins is the optimal hyperplane.[15]

2.2.3 Fuzzy Methods

Fuzzy logic is a technique that makes it easier to analyse system uncertainties when the uncertainty arises from the delay or "fuzziness" of the data. Fuzzy logic is a great set of traditional Boolean logic and has been expanded to include partial membership. A fuzzy set is a generalization of a common subset, e.g., B. a sharp subgroup, where the last membership function has only two values - 0 (full non-membership) or 1 (full membership). The Fuzzy Logic (FLS) system creates a series of fuzzy logic rules and membership functions that map the input vector (pointed inputs) to the scalar output (pointed outputs). FLS consists of three elements:

- Fuzzifier links net numbers with fog phrases and is required to activate the rules.
- The inference mechanism activates the rules as a series of expressions.
- IF-THEN - Fuzzifier produces sharp outputs based on the results of the fog.[16]

2.2.4 Generic Algorithm

The Genetic Algorithm (GA) mimics biological evolution. Each individual in the population should be described as a chromosome consisting of a series of genes from a particular alphabet. The alphabet, in turn, can consist of binary numbers, values, integers, symbols, matrices, and so on. The presentation determines the structure of the problem in

GA and defines the genetic operators used. For example, if a vector represents a chromosome with constant values, the length of the chromosome is the length of the solution vector, which are the coefficients of the model.

In GA, the search begins with an initial set of random solutions called a population. The chromosome of each population is evaluated by measuring capacity, which indicates the success of the chromosome. Based on the values of the compatibility functions, a set of reproductive chromosomes is selected. Genetic operators such as crosses and mutations are used to simulate the new generation. Parents and descendants are selected based on benefit assessments, some of which are excluded to keep the new generation population constant. The evaluation-selection-reproduction cycle continues until an optimal or near-optimal solution is found.[17]

2.3 Ensemble Methods

The ensemble methods combine predictions from multiple base estimates using a specific training algorithm to improve the generalization of a single estimate. They can be divided into two categories; average methods. The main idea is to create different estimators independently and then average their predictions and extension methods. The guiding principle is to combine several weak models to obtain all-powerful ones. Examples of the first category are bags and RF(Random Forest) methods, while the second category includes Adaboost and XGboost.

1) Technical characteristics: Electric charge data follows a pair of time series (time, value) and does not provide specific attributes for use with our predictive models. So, we had to use the previously available data to generate the features and use it as input for the models.
2) Feature Selection: Feature selection can sometimes be a useful technique for improving the accuracy of a predictive model. This largely depends on the correlation between features and target values and the feature selection methods used. Trying to reduce the number of independent variables used by the model without sacrificing accuracy is an important step in the modelling process. We performed the experiments using two feature selection methods. A detailed explanation of these methods would be beyond the scope of this project.

2.4 Benefit of Ensemble Model

The ensemble is predictive models that combine the predictions of two or more models. Ensemble learning methods are also popular and transition techniques when the best outcome of a predictive modelling project is the most important outcome. However, this is not always the most appropriate technique, and for those who are new to machine learning, it is expected that ranges or a particular method will always be the best method.

This model offers two project-specific predictive modelling benefits. It is important to know what those benefits are and how to measure them so that using the assembly is the right decision for this project.

The minimal benefit of using range is to reduce the variance of the mean skill of the prediction model. The main advantage of using gangs is the improvement of the average betting result of gang members. The mechanism for improving performance with ranges is often to reduce the variance of prediction errors made by contributing models.

The predictions can be combined using statistics, such as modalities or medium or more sophisticated methods, that teach how much and under what conditions each member can be trusted.

Research into ensemble methods appeared in the 1990s, and during that decade, articles were published on the most popular and commonly used methods, such as core attachment, reinforcement, and stacking.

By the end of the 2000s, overall adoption increased due to their tremendous success in machine learning competitions, such as the Netflix Prize and subsequent Kaggle competitions.

There are two main reasons ensemble is used in one model, and they are interrelated:

• Performance: The kit can provide better predictions and perform better than any assistive model.

• Strength: Overall reduces the propagation or dispersion of prediction and model performance.

2.4.1 Bagging

Bagging is utilized when the objective is to diminish the difference of a choice tree classifier. Here the goal is to make a few subsets of information from preparing tests picked arbitrarily with substitution. Every assortment of subset information is utilized to prepare their choice trees. Therefore, we get a troupe of various models. Normal of the relative multitude of forecasts from various trees are utilized, which is more powerful than a solitary choice tree classifier.

Stowing Steps:

Assume there are N perceptions and M highlights in preparing the informational collection. An example from preparing informational collection is taken arbitrarily with substitution.

A subset of M highlights are chosen arbitrarily, and whichever highlight gives the best split is utilized to part the hub iteratively.

The tree is developed to the biggest.

Above advances are rehashed n times, and expectation is given on the collection of forecasts from n number of trees.

Benefits:

- Diminishes over-fitting of the model.
- Handles higher dimensionality information quite well.
- Keeps up precision for missing information.

Burdens:

Since a definite forecast depends on the mean expectations from subset trees, it will not give exact qualities for the order and relapse model.

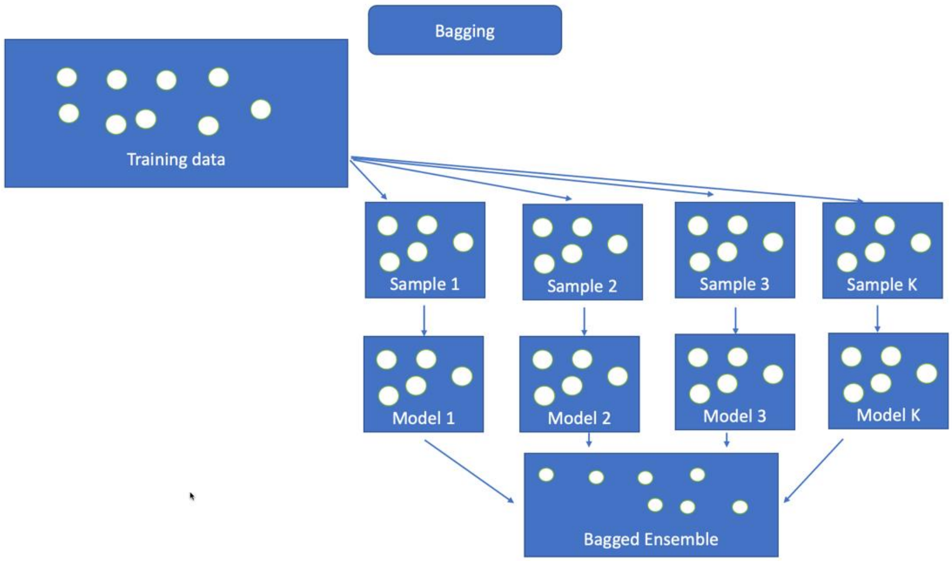

Figure 2. Bagging in Ensemble model

In bagging are use bootstrap sampling to obtain subsets of data to form a set of initial models. Bootstrap sampling is a process that uses ever-increasing random samples until declining yields of predictable precision are achieved. Each sample is used to form a separate decision tree, and the results of each model are summed. For classification activities, each model evaluates the result. In regression problems, the model score is calculated as an average. Low-budget but high dispersion models are suitable for bagging.[19]

2.4.2 Boosting

Boosting is likewise a homogeneous feeble students' model. However, it works uniquely in contrast to Bagging. In this model, students adapt successively and adaptively to improve model forecasts of a learning calculation.
XGboost: XGBoost is a decision tree-based troupe Machine Learning calculation that utilizes a slope boosting structure. In forecast issues, including unstructured information (pictures, text, and so on), fake neural organizations will, in general, beat any remaining calculations or systems.

XGBoost is an upswing AI computation in time plan illustrating. XGBoost (Extreme Gradient Boosting) is an overseen learning computation reliant upon boosting tree models. Such computations can explain how associations among features and target factors which is what we have anticipated. We will endeavour this system for our time plan data, most importantly, explain the mathematical establishment of the associated tree model.[18]

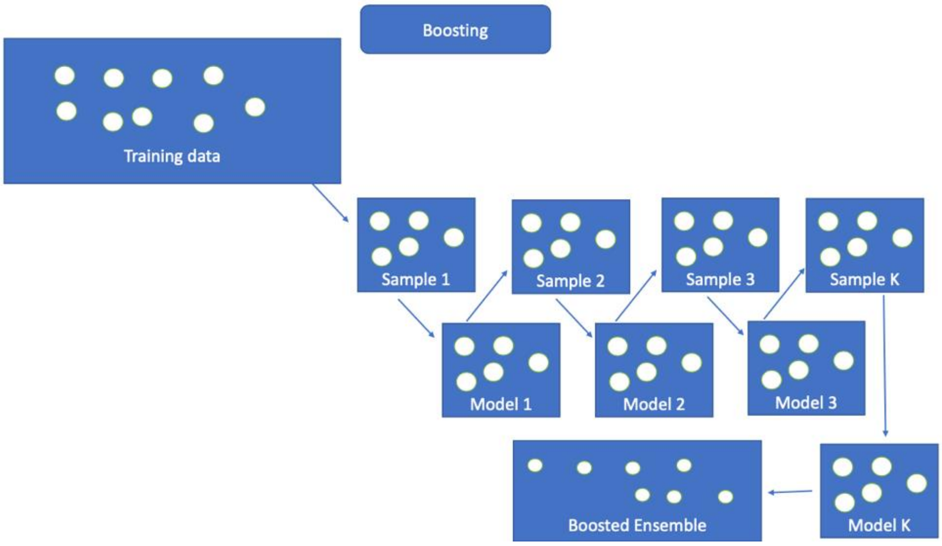

Figure 3. Gradient Boosting model

For boosting, we improve performance by focusing on data that generates more errors (i.e., focusing on tough stuff). We train several models where more weight gives examples that were misclassified in previous iterations. As in the case of bags, the classification operations are solved by a weighted majority. The regression operations by a weighted sum for a final forecast. Basic models with little variation but high bias are well suited to winning.[19]

2.4.3 Data mining Ensemble Model through XGBoosting

Boosting was introduced for numerical predictive activity. The overall model consists of core analytical designed using Long Short-Term Memory (LSTM). In boosting, this model works with XGboost, as there is a series of trains in this model for which the error in the test data is low. This extreme gradient boosting gain provides an efficient implementation of gradient models, which can be configured to drive random predictions in test data. Every model is used to make a prediction, and the actual total load forecast calculates the average of the predictions.

Random prediction is a simpler algorithm than gradient enhancement. The XGBoost library enables model training to replicates the computational efficiencies implemented in the library and uses them to train random forecast models. This sequential analysis of the XGBoost API is used to train and evaluate randomly predicted ensemble models for classification and regression in the ensemble model. The ensemble model matches the hyperparameters of the XGBoost random ensemble prediction model.

Gradient gain is the best choice for algorithms used to predict classification and regression design because it often offers the best performance. The problem with gradient enhancement is that training the model is often very slow, and large amounts of data exacerbate the problem. XGBoost solves the speed gradient gain problems by introducing several techniques that greatly speed up model training and often result in better overall model performance.[20]

2.4.4 Difference From Existing Models

This new ensemble model is calculating the values of the actual total load forecast in different countries differently. The actual total load has two different data types, which are before lockdown and during the lockdown. This section presents the experimental results of ensemble models trained on a specific overlapped subset of the training data. The structure of this experimental setup is described as follows: Every ensemble of this group consists of a different number of LSTM models and neurons in each layer. Training the ensemble models on overlapped subsets of the data performs less when compared with a model trained on the entire dataset. For this project, we have data from December 2019 to May 2020. In this project, we are working on hourly, Weekly, Weekend, and monthly. Through this different calculation, we can observe that before lockdown train and test data difference and same observation in during lockdown time.

In this project, we use Extreme Gradient boosting because there is less difference between training data and test data. These methods for predicting a long period of pandemic time find out the actual error. The ensemble xgboost model works sequentially so it can get a suitable model from other existing models.

The goal is not to get the best possible model for each time series but to avoid bad enough models. In other words, making small mistakes every time series is not a problem, but making the small error, every model has a little bit—the thought it could do it by combining different techniques using calculated models.

This means that while XGBoost is the best approach for a particular series, it may not be the best for another series. The same is true for exponential smoothing. However, if the combine a model from each technique, even if one model is not very good, the other brings the estimate closer to the actual value. XGboost works best for long-term, well-behaved series, while exponential smoothing occurs for short-term noisy series.[29]

2.4.5 Methods for Evaluation Metrics

Different statistical metrics is giving different types of value in before lockdown and during lockdown. In this project have total 3 different metrics. Loss function is an important components of Neural Networks.

Loss Function: It's a component of Neural Networks. Loss is a Neural Net prediction error and a loss calculation method. Simply put, the calculation of loss gradients. It is an integral part of neural networks. This is how a Neural Net is trained. Keras and Tensorflow have various inbuilt loss functions for different objectives. In this project, I covered the following essential loss functions, which could be used for most of the objectives. [31]

RMSE: The RMSE is the Standard Deviation of the predicted error. The remaining components measure the distance of the data points from the regression line; RMSE is a measure of the distribution of these residues. In other words, it tells us how much data is concentrated around the most appropriate line. The root means the square error is commonly used in climatology, prediction, and regression analysis to verify experimental results.[31]

STD: In statistics, using the Standard Deviation is helped to understand the distribution of all data many bits better. The STD is the measurement of the spread out numbers are. Using the Standard Deviation, we have a "standard" way of knowing what is normal and what is extra-large or extra small. Without measuring the standard deviation, we cannot handle whether the data are close to the average.[31]

Mean Absolute Percentage Error: The MAPE is a measure of how accurate a forecast system is. It measures this precision as a section and can be calculated as the average absolute percent error for each period minus actual values divided by actual values. MAPE is normally used for regression problems and model evaluation because of its intuitive interpretation in terms of relative error.[31]

2.5 Method of Data analysis and Feature extraction

2.5.1 Hourly

Forecasts of individual households' average daily energy consumption for the next day are important for many real applications. For example, daily household energy forecasts can support wholesale electricity household offerings. The load on the utility units can provide a fixed demand-response curve based on knowledge of energy consumption. Daily average of each user on the next day. The average daily energy consumption can also be useful for storing energy in relatively small systems and electricity from the archipelago to the family level. This forecasting problem was also addressed in a recent assessment of energy management during a pandemic.

2.5.2 Weekly

Load forecasting is also done weekly. From this differentiation in the project, we understand the weekly electricity consumption. During this pandemic, we can also see how this electricity consumption is developing. In each season, data from one week is selected as a set of ENTSO data and data from the next day as a series of data for validation. The forecast for next week was chosen as the case for the long-term forecast to examine the image. In this study, the experiments were performed over six months, using data from the last three months and after lockdown the data as a training set. The following week data as data sets for validation.

2.5.3 Weekend

Predicting maximum load prevents energy loss and helps solve long-term load forecasting problems by planning electricity consumption. In other words, the peak load forecast should be used to determine energy policy as part of the simulation data. From the operator's point of view, it is possible to manage installations without problems such as power outages and reduce costs by providing a stable power supply for the system. Also, the predicted results can be compared to actual readings to use a system security assessment to pre-empt various vulnerabilities. Therefore, various studies are constantly performed to predict peak loads reliably. Most of these studies focus on extending existing predictive models or developing new ones. This removes variables that are not suitable for prediction by selecting or allocating new features to improve accuracy. Thus, good prediction results can be obtained if the model offered in these search results is trained with the appropriate data. Recently, various combinations of variables that can be used to predict peak load have been studied. Training methods are being developed so that the accuracy of peak load forecasting is gradually increasing.

2.5.4 Monthly

Month to month electric burden essential is fundamental to work metropolitan power grid proficiently. Although different determining models dependent on human-made reasoning strategies have been proposed with proper execution, they require adequate preparation datasets. On account of month-to-month anticipating, because only one information point is created each month, it is not easy to gather adequate information to develop models. This absence of information can be reduced utilizing move learning procedures. This paper proposes a novel month-to-month electric burden anticipating plan for a city or region dependent on the move taking in utilizing comparable information from different urban communities—a few estimating models dependent on profound learning approaches month-to-month energy utilization forecast. The good month-to-month power utilization information is trying because of the long chronicle time frame; it is hard to construct a complex determining model dependent on an AI strategy.

2.6 Research Methodology Tools

In this thesis, research of quantitative data is conducted. More precisely, the analysis of time series data is envisaged. When working with time-series data, various methods are used to acquire data parameters and functions and prepare a forecast. Among the methods listed in Chapter 2, current work focuses on statistical methods, e.g., linear regression, ML model and Ensemble model. The research method is research because the development of functions tries to express the properties of the data, the relationships of the variables, etc. When predicting time series, normal operations must be performed.

- problem definition,

- description and classification of data,
- research analysis by examining data properties using models, converting data as needed, followed by correlation analysis,
- Selection and adaptation of variables to the linear regression model based on the previous analysis.
- use and evaluation of forecasting models through training and validation kits.
- Analizing all data we used ENTSOs data and there actual total load value in hourly.
- In this thesis the software aspect is Anaconda.nevigator, Jupyter Notebook and python libraries.

```
import numpy as np
import pandas as pd
from sklearn.metrics import mean_absolute_error
from matplotlib import pyplot as plt
from xgboost import XGBRegressor
from sklearn.model_selection import cross_val_score
from sklearn.metrics import median_absolute_error, mean_absolute_error
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.preprocessing import MinMaxScaler
import warnings
warnings.filterwarnings("ignore")
```

```
#Importing liberaries
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
import math
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings("ignore")
```

Figure 4. Python jupyter notebook libraries for LSTM and XGBoost

- Those values are divided by hourly, weekly, weekend, and monthly. That data had before lockdown three months and during the lockdown three months. The data comparison will help to make an ensemble model through XGBoost.
- End of this analysis, deliver the results.

Based on the above, the research method is close to case analysis, where the difference between the analysed data is quantitative rather than qualitative.

3 Numeric Data Analysis

After analysing the sources and datasets, a function design is performed that includes the following steps:

- The properties of the data become apparent by examining them in graphs. For each variable 2019 and 2020, a graph explained and predicted variables should be provided to ensure a linear model between the type of relationship and the explained and reactive variables, where transformations are required, and Transformation is needed. Possible conversion options are described.
- Correlation analysis is performed to measure the relationship between responses and explanatory variables. Indicates which variables should be maintained for model development.
- Each variable is assessed separately by completing a template and examining its summary and performance.
- Based on the output variable, it is used to create the final model and validate the variables that affect the model's performance and improve the LSTM, RMSE, Loss function and MAPE based on the output variable.
- The model is tested, and the average set is calculated between 01.12.2019 and 31.05.2020.

In Estonia, The comparison of data before lockdown and during the lockdown time. Through this comparison, this during lockdown uses data is decreasing.

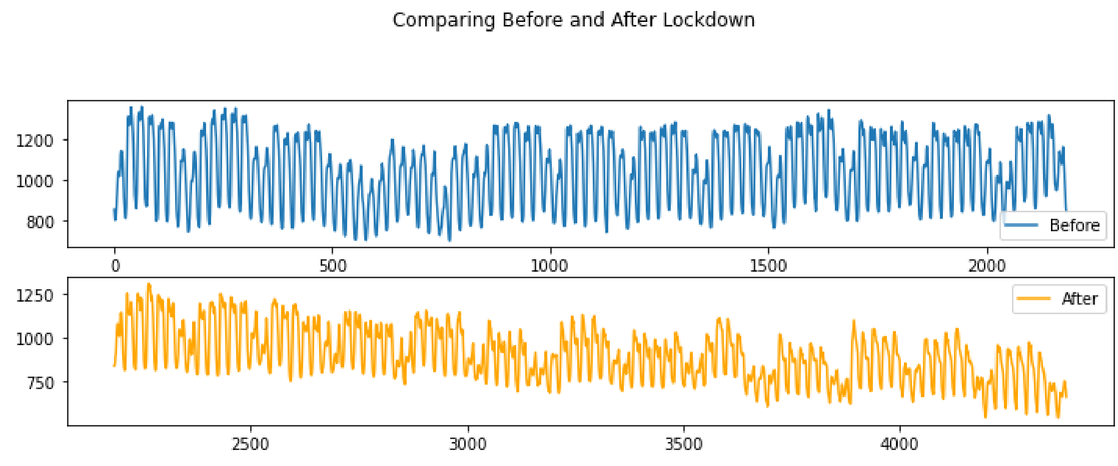

Figure 5. Compare before and during lockdown for Estonia

For Finland, the same period time of data has to compare to analyse the difference and get a result. In this comparison show how much change came in this two different time. In Finland, the use was high before lockdown, and the use was low during the pandemic time.

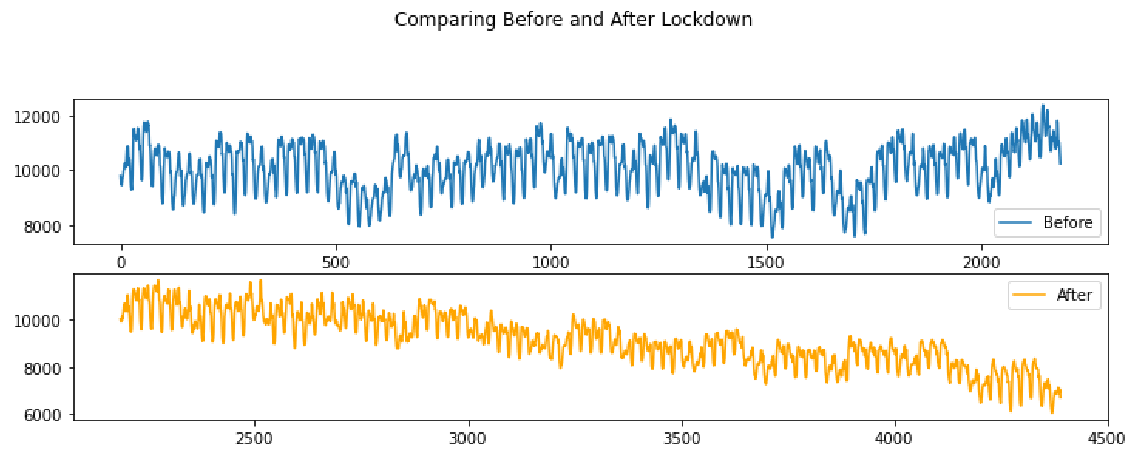

Figure 6. Compare before and during lockdown for Finland

Norway had more changes than other two counties because this comparing graph is show the uses in during lockdown was decreased so sharply. In this graph mainly comparison of that before lockdown and during lockdown time.

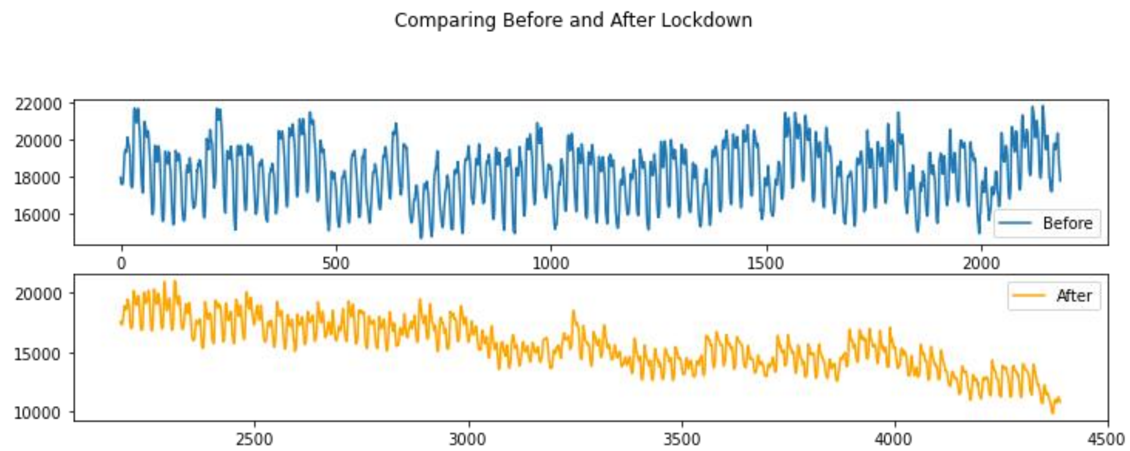

Figure 7. Compare before and during lockdown for Norway

After examining the innumerable literature on short-term electric uses prediction, selection of prediction methods, and algorithms, it can be concluded that the basic rule is which prediction technique should be used in a specific situation that other methods can surpass.

The ensemble model developed during the initial project forecast in electrical load in a short time in one day with hourly, weekly, weekend and monthly accuracy. The uses of total actual load data is available. NB! For technical reasons, total actual load of Estonia, Finland and Norway consumption are available from December 2019 to end of May 2020.

In this project, we work on a total of six months of data for three different countries(Estonia, Finland and Norway). Then splitting the dataset into train and test split using scaler.fit_transform() method. To convert an array, we need to import numpy and make dataset metrics. The data preparation and splitting processes are given below:

- Training Dataset: This dataset is used to train the model. This data set should be the largest portion of the data set.
- Validation Dataset: If they require several test models, and the validation dataset is used to test models in all of them.
- Test Dataset: In which way, the best model selected through the validation performance performs on a third, and for the model, previously with the unknown dataset.

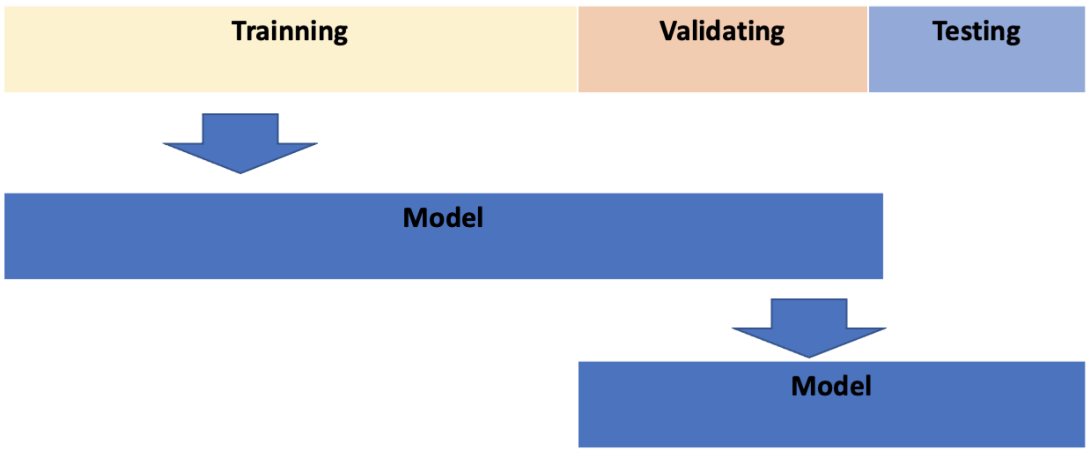

Figure 8. Multiple Model Validation

At first, through this model validation, build a model based on data that is not supposed to be known. The second step is the overfitting process of designing a model adapted so closely to historical data, and it will be ineffective in the future—lastly, the underfitting for prediction.

If the specified value of x is outside the minimum and maximum values, the resulting value will not be between 0 and 1. We can review these views before making predictions and removing them from the dataset or limiting them to maximum values or predefined minimums. We can normalize all data set using the Scikit MinMaxScaler learning object. The following are best practices for using MinMaxScaler and other scaling methods:

• Adjust the scale based on the available training data. For normalization, this means that training data are used to estimate the minimum and maximum observed values. This is accomplished by calling the fit () function.

• Apply the scale to the training data. This means that we can use standardized knowledge to train the model. This is accomplished by calling the transform () function.

• Apply to out-of-scale data. This means that we can produce the new data that we want to predict in the future.

The default scale of MinMaxScaler is to change the scale of variables in the range [0,1]. However, the desired scale can be specified using the "feature_range" argument, and a set of minimum and maximum can be specified for all variables.[30]

This ensemble mode technique is constructing a prediction model. The choice of this method is based on the information provided on LSTM and XGBoost. The ensemble XGBoost method is suitable for working with time-series data, measuring the relationships between variables, and managing many error of prediction. Unlike time series methods, e.g., ML models in which the estimation of future data is based on historical data models, predicting them for the future, regression models are relevant when historical time series data are available. However, some variables affect the predicted load and are included in the selected forecast. Actual and predicted property values can be added to the model calculation.

3.1 Prediction Process

Depending on the frequency of observations, the time series can generally be the hour, day, week, and month. Sometimes it can be in seconds and minutes as total power consumption uses. Time series analysis involves understanding several aspects of the nature of the series to be better informed about meaningful and accurate predictions.

We are using a read_csv () time series dataset as the 'pandas' data frame in the panda package. Adding the dataset.describe().transpose() command will make the date column be parsed as a date field.

Alternatively, the import it as an array of pandas with the date as an index. All have to do is specify the index_col argument in pd.read_csv (). Each time series can be divided into the following components: baseline + trend + seasonality + errors.

The trend is observed when an increasing or decreasing slope is observed in the time series. At the same time, seasonality is observed when a different pattern of repetition is observed between seasonal factors due to seasonal factors. It could be the month of the year, the day of the month, the days of the week and even the time of day.
However, all-time series do not need to show trends and seasonality. Time series may not have a clear trend, but they do have seasonality. It could be the other way around. Typically, panel data columns contain explanatory variables that can be useful for predicting Y, provided those columns are available in a future forecast period.

Using matplotlib visualize the series. Then splitting the dataset into train and test split. To convert an array, you need to import numpy and make dataset metrics. Next step is reshare like X=t,t+1,t+2,t+3 and Y=t+4. Later it will be a stacked LSTM model. Lastly, with check prediction using the following functions:
ori_train_predict=model.predict(X_train)
ori_test_predict=model.predict(X_test) [32]

3.2 LSTM Numerical Analysis

In the thesis, the ensemble assumes that the relationship between train and test data through XGboost. Fortunately, small deviations from this model do not affect many regression procedures. However, suppose the curvature of the relationship between the predictor and the predicted variables is visible from the scatter plot. In that case, this research project may need to consider converting the variables or enable explicit nonlinear components.

The final sections of the document summarize the work results, draw conclusions and provide possible guidelines for future work. The result of the work is discovering new functions and improving the existing model of the global model. At the end of this article, the inquiries referred to in the previous subparagraph should correspond to the required evidence.

This section describes the Numerical results of the proposed LSTM model for predicting consumption in Estonia, Finland, and Norway. Table 1 below describes the Numerical results of an experiment conducted using consumption data on hourly, weekly, and Monthly. Weekend features are also predicted and metrics. Such as STD, RMSE, Loss function, and MAPE. We are used to estimating the performance of the model based on the electricity consumption data.

3.3 Estonia

In Estonia, load forecasting for any pandemic and before pandemic time analysis through the train and test data. This graph can help understand how much error or loss have been in normal time and in covid-19 time from this data analysis.

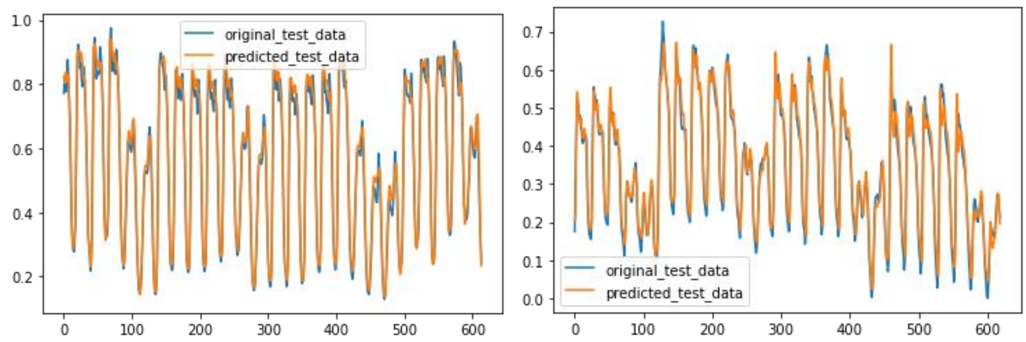

Figure 9. Predicted and original prediction for Estonia in before and during lockdown

The experimental is showing the difference between before lockdown original test data and predicted test data. Another graph is also showing the same thing in during lockdown time. By comparing all this data consumption uses behaviour is changes in during lockdown time and it’s decreasing. The x-axis is the main index and y-axis is the original data in blue colour. The orange colour of line is the predicted data.

3.3.1 Root Means Square Error in Estonia

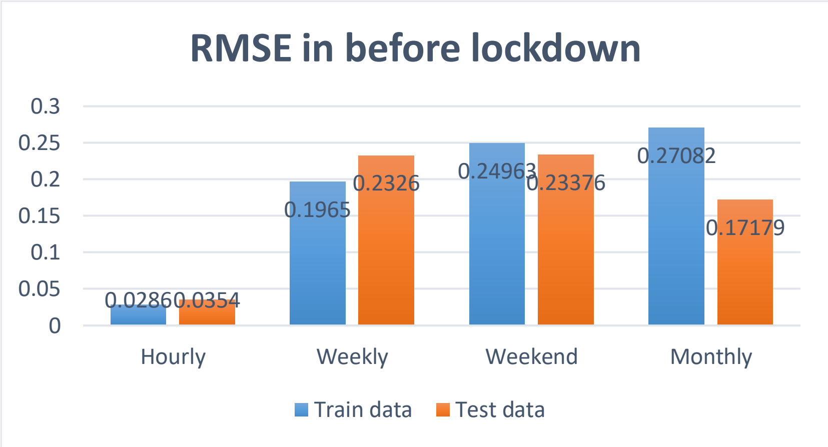

Figure 10. RMSE in before lockdown in Estonia

Figure 9 is describing the analysis of the root means square error before the Covid-19 pandemic in Estonia. Analysis shows that RMSE train data is larger in the Weekend and Monthly sectors where test error is lower than Hourly and Weekly. On the other hand, hourly and weekly test error prediction is bigger than train data during the pandemic time. Intuitively suggesting that there was more error recorded hourly and weekly before the pandemic December 2019 to February 2020.

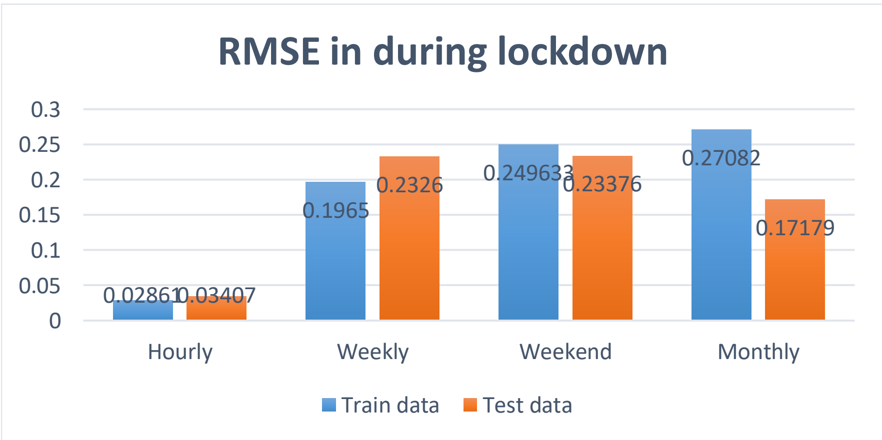

Figure 11. RMSE in during lockdown in Estonia

Figure 10 illustrates the root means a square error during lockdown between March, April, and May 2020 in Covid-19 time. From this bar chat, it can be seen that the number of errors is increased hourly and weekly. However, on the weekend and Monthly, it was relatively low during the pandemic time. Overall we can see that weekend and monthly error was low in both times.

3.3.2 Standard Deviation in Estonia

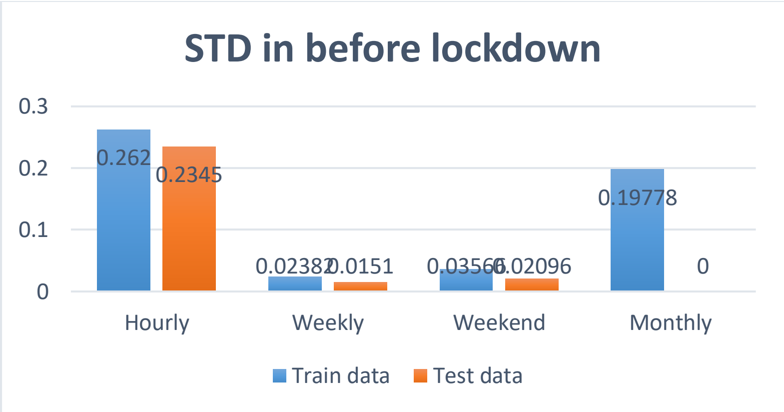

Figure 12. STD in before lockdown time in Estonia

Figure 11 is describing the analysis of the standard deviation before the Covid-19 pandemic in Estonia. This analysis shows that STD train data is bigger in hourly, week by week, end of the week, and month to month, where test data error was low. In any case, the monthly test data expectation was 0, where the train data error was 0.19778 during the pandemic time. Before the pandemic from December 2019 to February 2020, general test data was low.

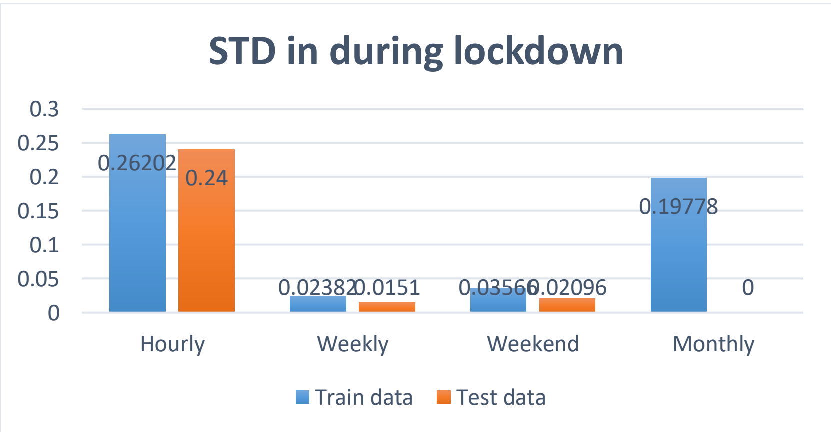

Figure 13. STD in during lockdown time in Estonia

The sample mean's standard error is the standard deviation of the set of means that would be found by drawing an infinite number of repeated samples from the train data and test data for each sample. From this analysis, we can understand that our standard test data error is low during lockdown time, where train data is higher in every portion.

3.3.3 Mean Absolute Percentage Error Estonia

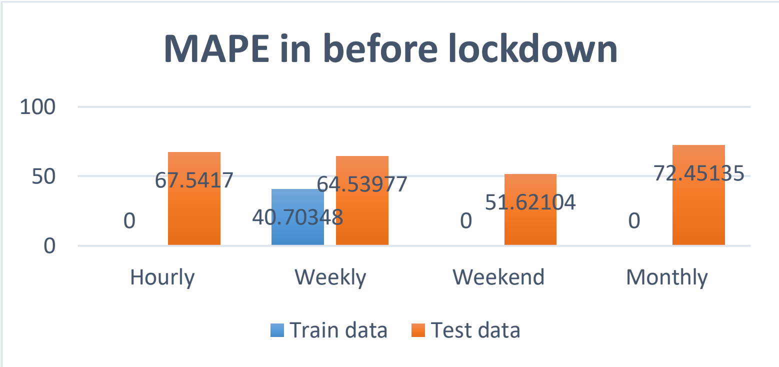

Figure 14. MAPE in before lockdown time in Estonia

The Means Absolute Percentage Error will also be calculated before and during the pandemic time data to ensure that the model predicts high accuracy. MAPE can calculate the accuracy of our forecast. This is an important part of our report because we rely heavily on future estimates and are making active predictions of future data sets, thus giving ourselves and other constituents the ability to know the accuracy of the prediction is paramount. The calculation is done by taking the difference between the actual total load value and dividing the difference between tarin data and test data.

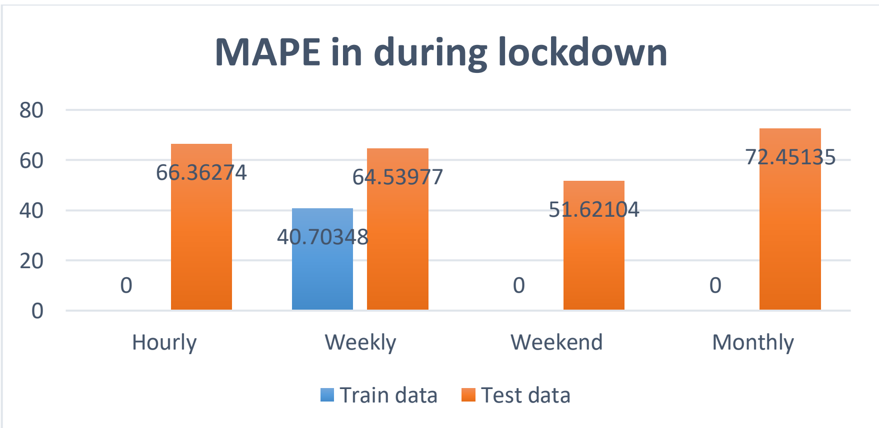

Figure 15. MAPE in during lockdown time in Estonia

This means absolute percentage error have similarity in before pandemic time and during the pandemic time. In that graph, we can see that only in weekly have train data error, but it dramatically increased for test data. However, in another sector, the test data error is always high.

3.3.4 Loss Function in Estonia

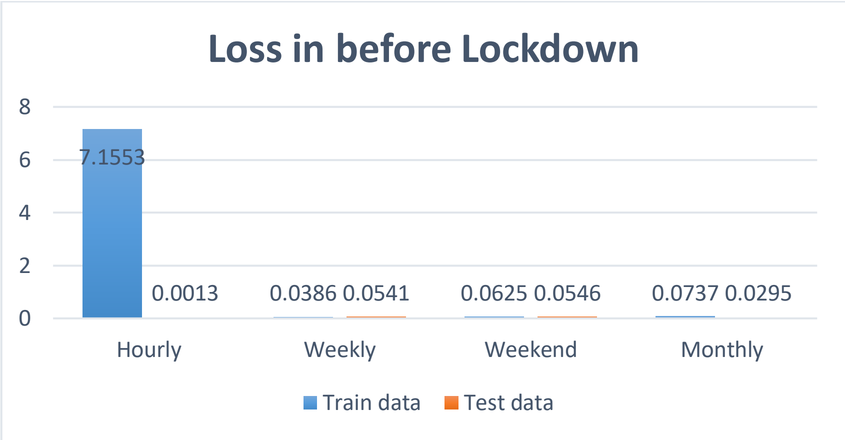

Figure 16. Loss in before lockdown time in Estonia

This graph shows that hourly train data is high, where test data is very low. The weekly, weekend and monthly loss functions are similar for train and test data. However, before lockdown, weekly, weekend, and monthly test loss functions are low than train data loss functions.

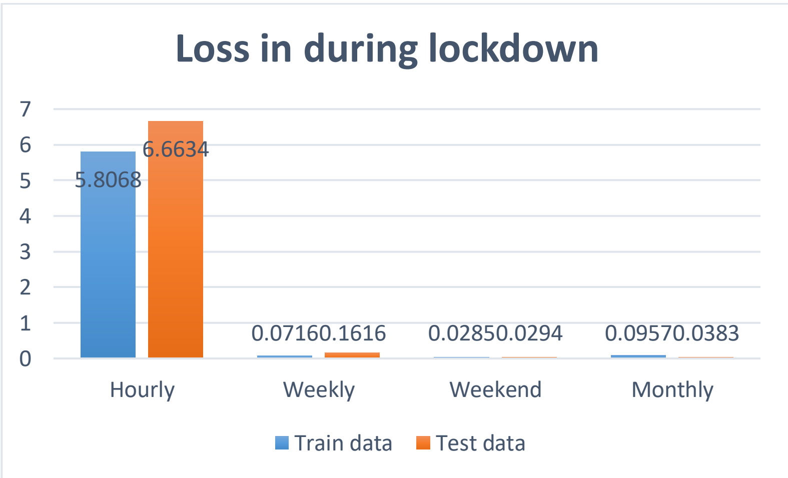

Figure 17. Loss in during lockdown time in Estonia

From this loss function analysis, we can see that the test loss function is high hourly during lockdown time, where the train loss function is small. Also, the weekly loss prediction error is high. However, On weekends and monthly, the prediction loss function error is lower than the train loss function.

3.4 Finland

This section describes the RMSE, STD, MAPE, and Loss function error which will help predict our next pandemic time for electrical energy load forecasting. Through this data analysis, the prediction will be understandable for any new pandemic time load forecasting by training data and test data.

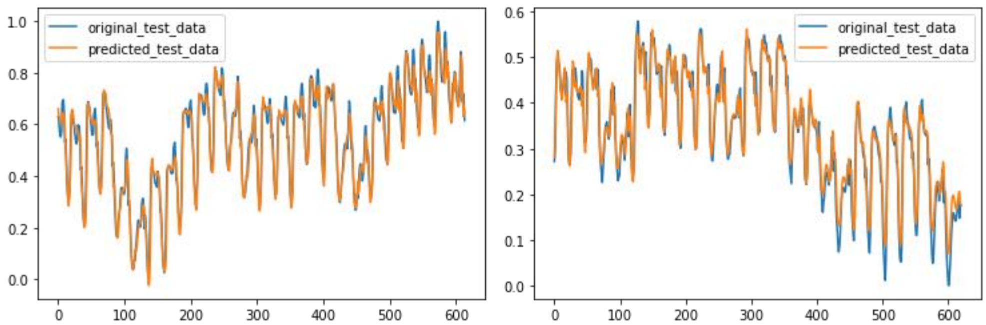

Figure 18. Predicted and original prediction for Finland in before and during lockdown

In descriptive research, that data is compareing the before lockdown and the during lockdown time. Before lockdown time the uses was high where in during lockdown time was low. From all of this graph are showing the effects on electricity.

3.4.1 Root means square error in Finland

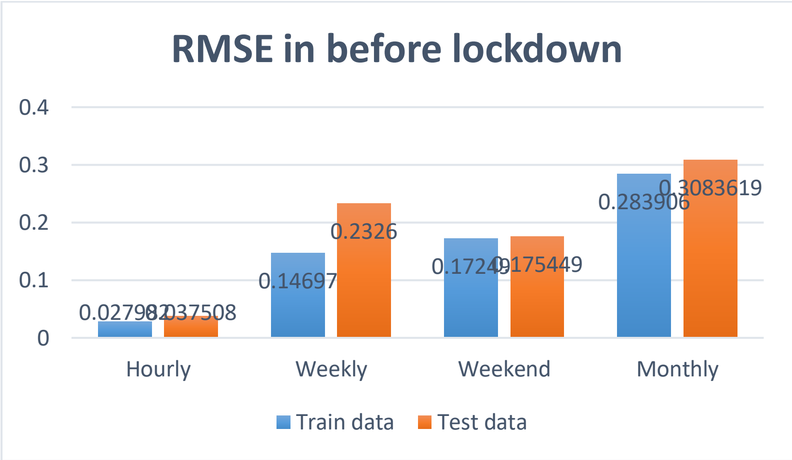

Figure 19. RMSE in before lockdown time in Finland

Figure 18 is describing the root means square error in before lockdown time through the train and test data. In this graph the test error is higher than train data error for hourly, weekly, weekend and monthly. However, for observation the train and test data have smaller difference error in hourly sector but in weekly have a big amount of error difference.

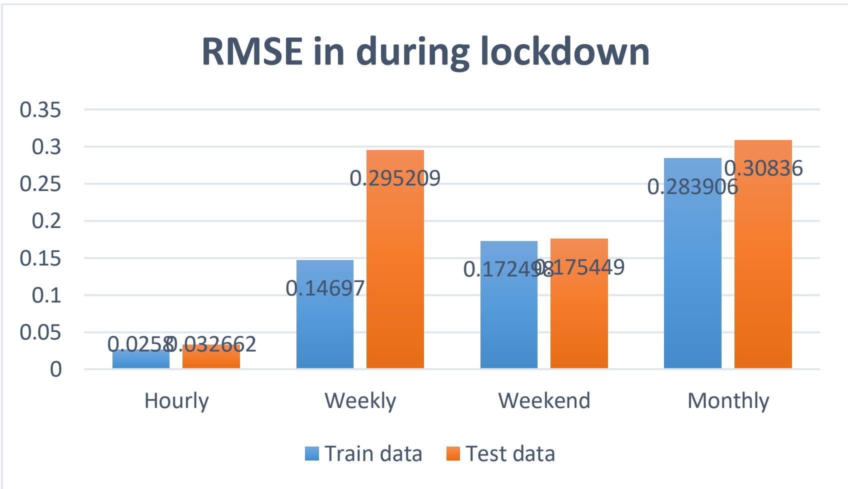

Figure 20. RMSE in during lockdown time in Finland

This figure is showing the train data and test data error beteween of them. In this graph is showing the prediction in during the lockdown time for energy load forecasting. This root means square error is showing the test data error and train data error and the weekly test data error prediction is higher than train data error.

3.4.2 Standard Deviation in Finland

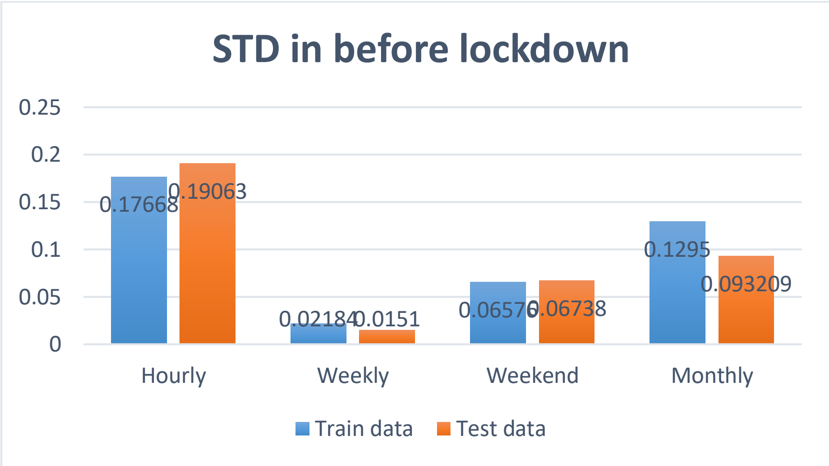

Figure 21. STD in before lockdown time in Finland

This graph is describing the Standard deviation before lockdown time. In this graph, test data is higher than train data, and this same data is almost similar in the weekend. The train data on the weekend is 0.06576, where the test data is 0.06738. However, in monthly prediction, the error is smaller than the train data.

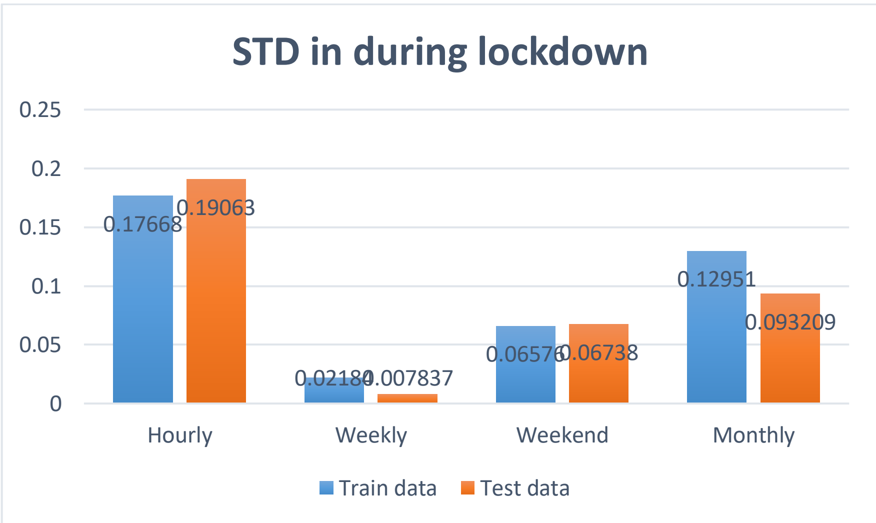

Figure 22. STD in during lockdown time in Finland

This bar chart is showing the difference between train and test data. From this graph analysis, the weekend is similar before lockdown and during the lockdown, but the changes are monthly. The prediction test error is smaller than before lockdown. On the other hand, it is the opposite of hourly purpose because this sector has big test data error and the train data error is smaller.

3.4.3 Means Absolute Percentage error in Finland

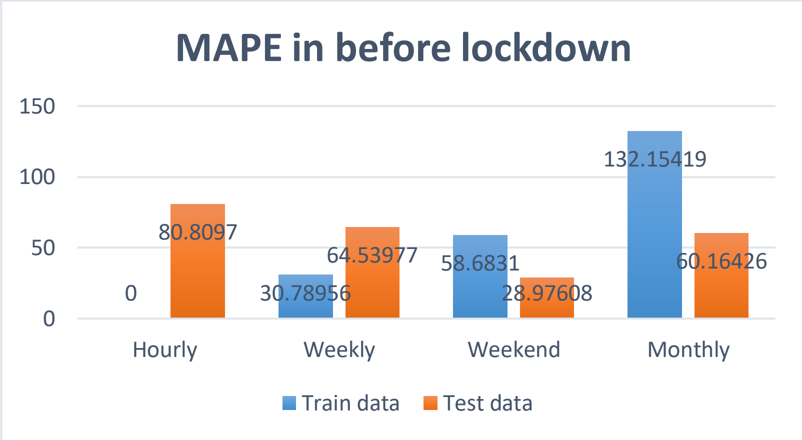

Figure 23. MAPE in before lockdown time in Finland

From these statistics, the means absolute percentage error explanation is before lockdown time. In this analysis, the hourly and weekly test error is higher than train error but oppositely the weekend, and the average monthly test data is smaller than train data. Before lockdown time, the monthly train data was 132.15419, where the test data was 60.16426, and the weekend also has low test error.

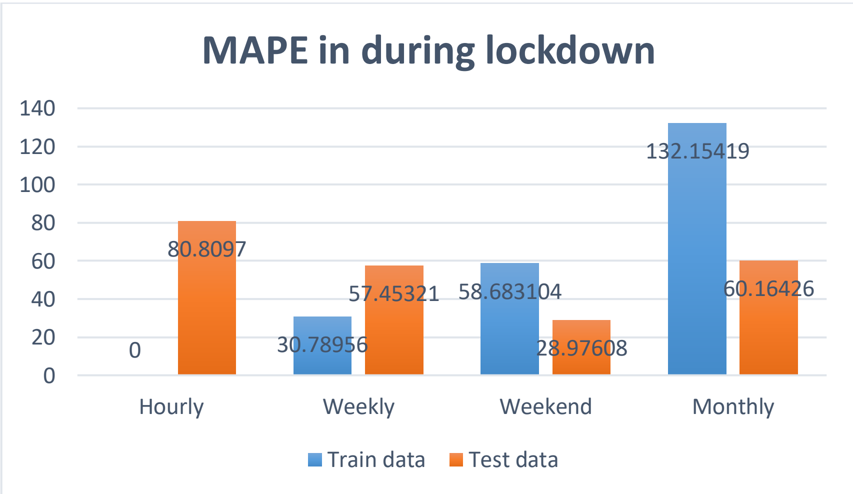

Figure 24. MAPE in during lockdown time in Finland

The during lockdown time, the means absolute percentage error is almost similar to before lockdown time. This graph has only one noticeable part for weekly where the test data error is decreasing in during lockdown time. However, the monthly and weekend are similar to the before graph.

3.4.4 Loss Function in Finland

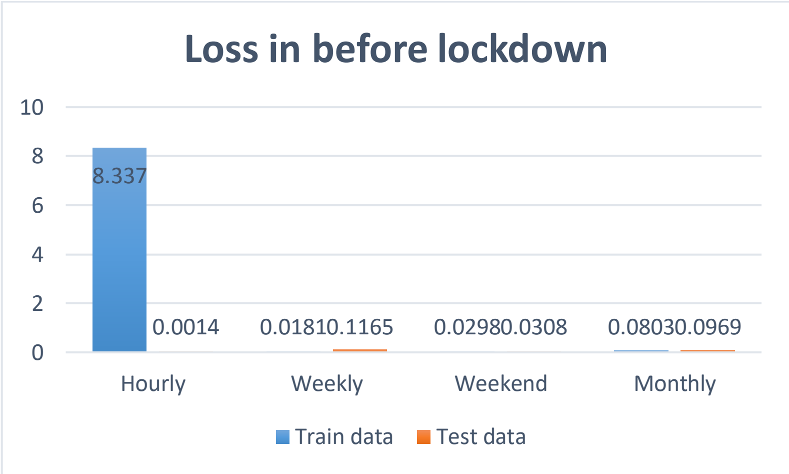

Figure 25. Loss function in before lockdown time in Finland

From this analysis, the graph is showing the loss function before lockdown time hourly, weekly, weekend, and monthly. When the hourly train loss is 8.337, then the prediction loss function is 0.0014, and from these hourly two values, the hourly error is smaller than another one. The weekly, weekend and monthly train and test loss function have a small error difference.

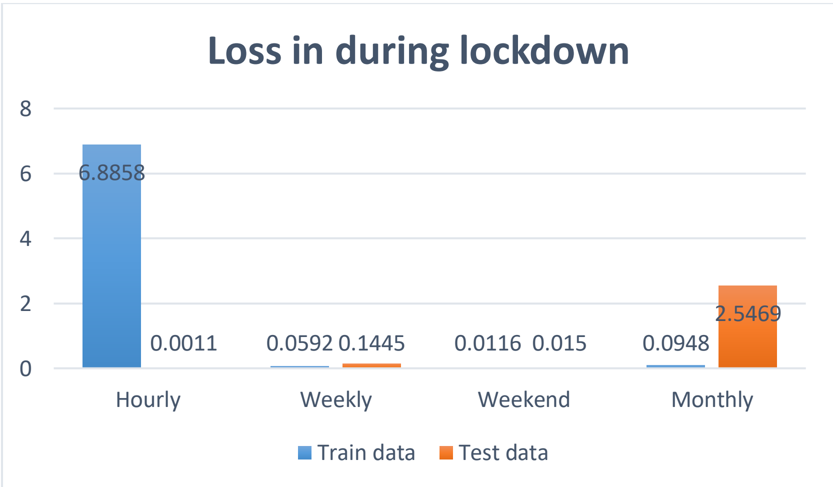

Figure 26. Loss in during lockdown time in Finland

Through this bar chart, the hourly train loss function is bigger than the test loss function. The train loss function is 6.8858, where the test loss function is 0.0011. On the other hand, it is fully opposite in monthly purpose because in monthly train loss function is 0.0948, and the test loss function error is bigger, like 2.5469.

3.5 Norway

In this time series prediction, for analysing the three countries' data through root means square error, means absolute percentage error, standard deviation, and loss function. So, these all methods are analysing by the hourly, weekly, weekend, and monthly.

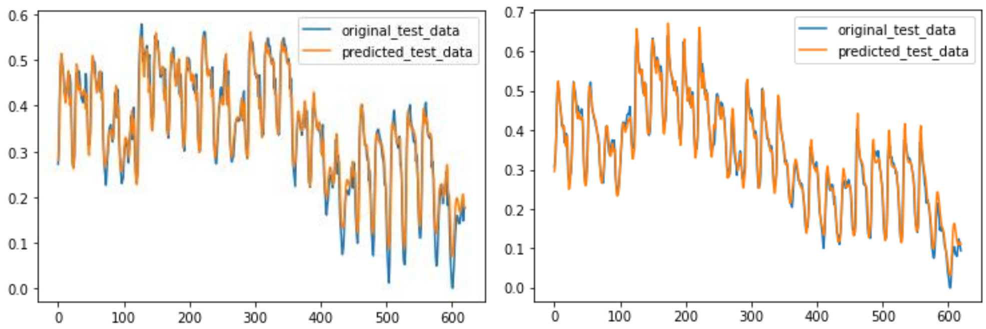

Figure 27. Predicted and original prediction for Norway in before and during lockdown

The graph presents the difference between before lockdown use and during lockdown time energy uses. Before the lockdown test, the original data and the during lockdown test data were different in Norway. This analysis of the graph also has a high level of predicted test data. The predicted test graph line is decreasing sharply.

3.5.1 Root Means Square Error in Norway

The RMSE is a quadratic scoring rule which measures the average magnitude of the error. The equation for the RMSE is given in both of the references. Expressing the formula in words, the difference between forecast and corresponding observed values are each squared and then averaged over the sample. Finally, the square root of the average is taken. Since the blunders are squared before they have arrived at the midpoint, the RMSE gives a generally high weight to huge mistakes. This implies that the RMSE is most helpful when huge blunders are especially bothersome.

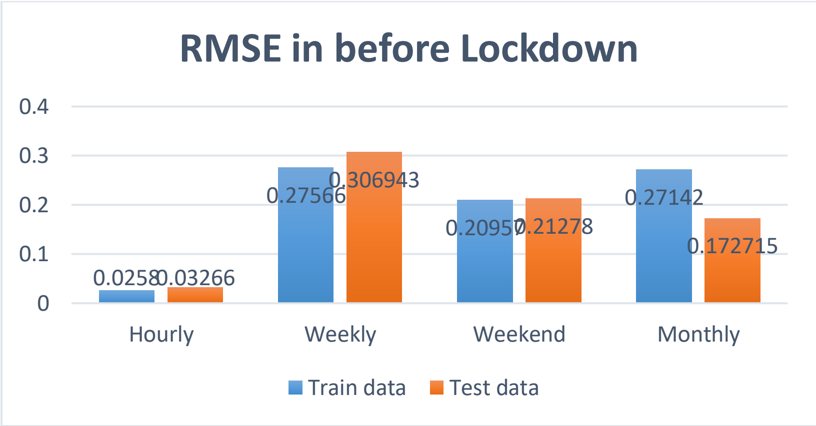

Figure 28. RMSE in before lockdown time in Norway

The root means the square error is explaining through the train and test value. In this bar chart, the monthly test prediction error is smaller than the other portion. There is no bigger error difference on hourly and weekend, but the weekly test error is bigger than the training value.

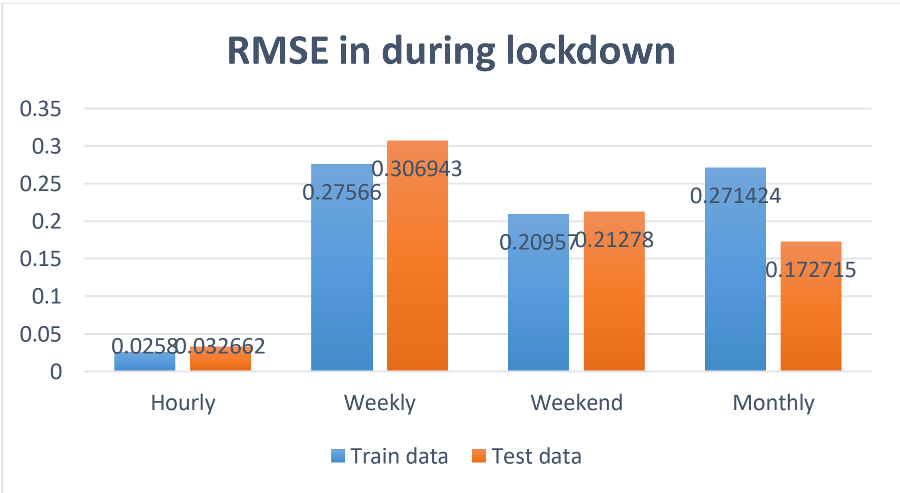

Figure 29. RMSE in during lockdown time in Norway

In this lockdown time also the same error in the train and the test value. However, the monthly root means a square error in the test sector was low, and monthly train values were 0.271424, where the monthly test value is 0.172715. From this graph, we can understand that Norway has the same situation before lockdown and during lockdown time. From these statistics, the monthly test prediction is small, and it is good for the model.

3.5.2 Standard Deviation in Norway

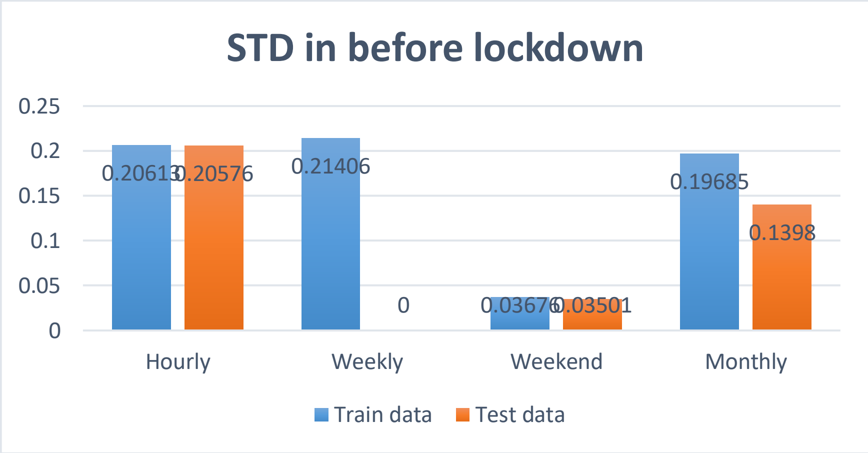

Figure 30. STD in before lockdown time in Norway

The Standard deviation is in before lockdown time is the same for hourly purposes. Secondly, the weekly test STD is 0. Thirdly the weekend and monthly test values are smaller than the train data value. The weekend test value is 0.03501, and the monthly test value is 0.1398, respectively, where the training value is 0.03676 and 0.19685.

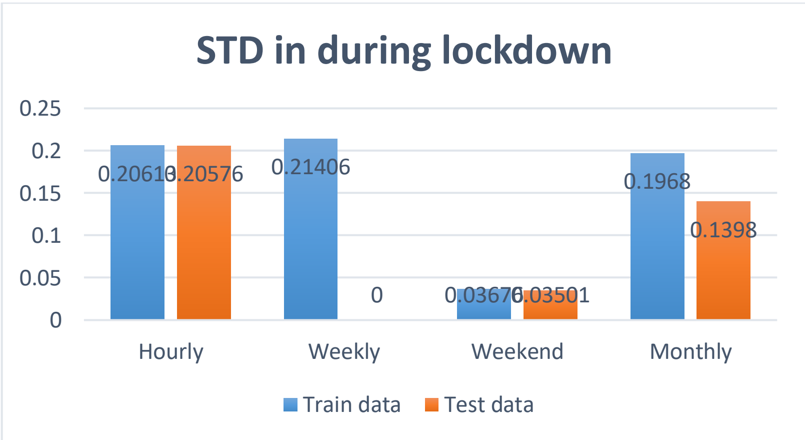

Figure 31. STD in during lockdown time in Norway

In this bar chat and before the lockdown time, bar chat also has a similar train and test value. Norway has not much difference between a lockdown and before lockdown time. From this analysis graph, the test value is suitable for prediction.

3.5.3 Means Absolute Percentage Error in Norway

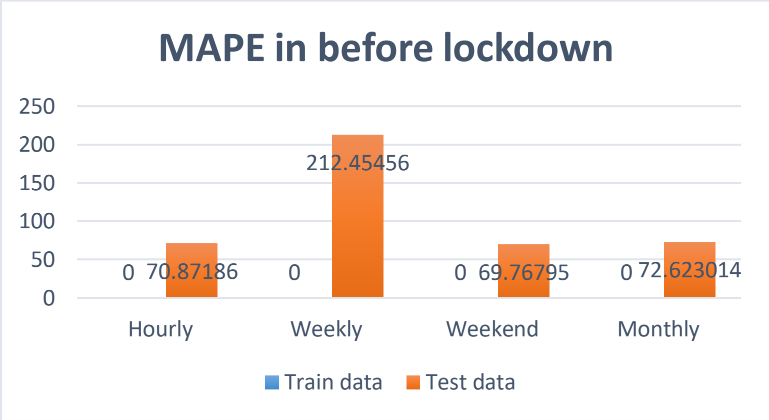

Figure 32. MAPE in before lockdown time in Norway

This bar chart shows the means absolute percentage error for Norway. Here the training data is infinitive, but the test data lower than train data. Overall in monthly test error is lower than train data error.

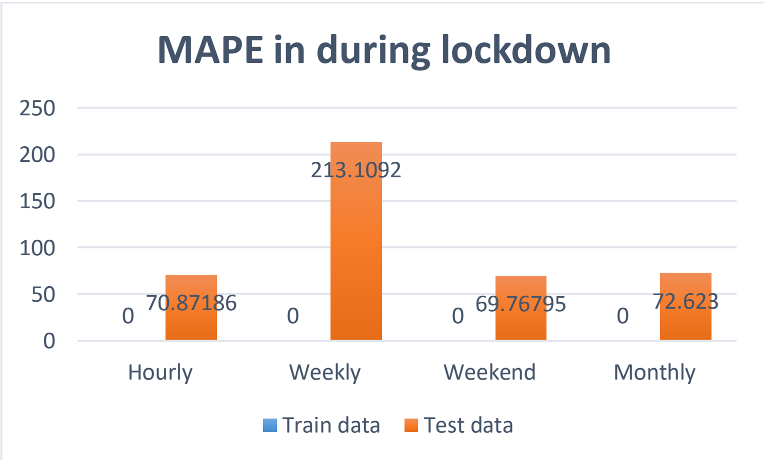

Figure 33. MAPE in during lockdown time in Norway

From this analysis, we can see that the before lockdown and during lockdown means absolute percentage error is similar. The monthly train data error is infinitive, where the test error is 72.623. Finally, we can understand through this graph that the means percentage error is acceptable for future prediction.

3.5.4 Loss Function in Norway

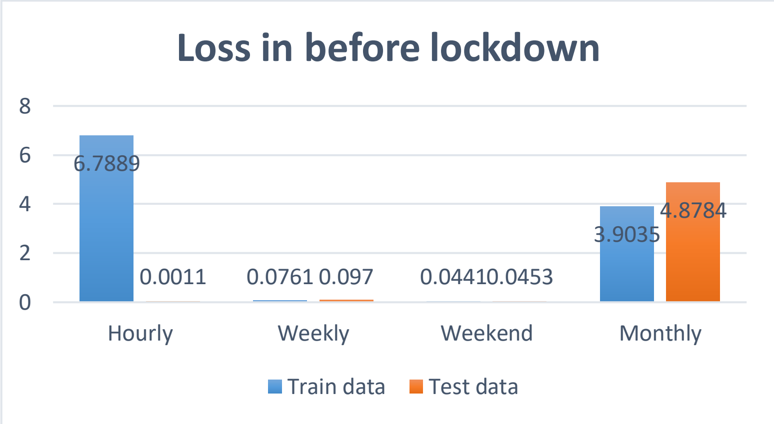

Figure 34. Loss in before lockdown time in Norway

This statistics graph shows the loss function before the lockdown time; the train loss function is higher than the test loss function hourly. On the other hand, we can observe that the monthly test loss function is smaller than the train loss function. In the weekly test, the loss function is different, and this test loss function is higher than another one.

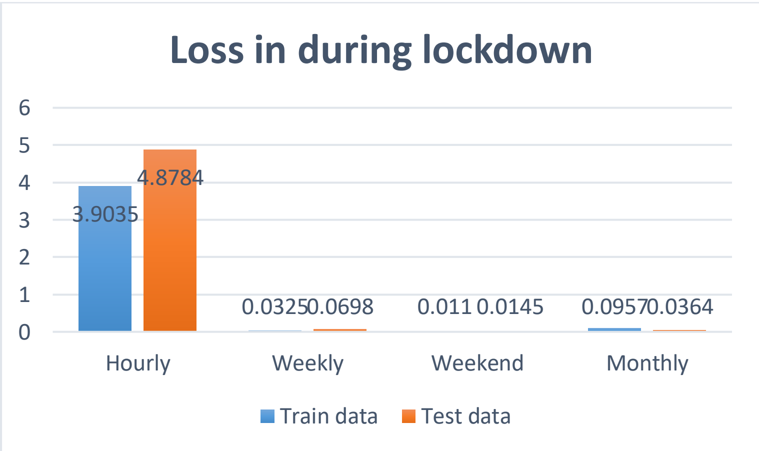

Figure 35. Loss in during lockdown time in Norway

From this statistics bar chart have less loss function in test data prediction. In monthly, the train data loss is 0.0957, and the test loss function is 0.0364. The loss function hourly has a difference between train and test where the test loss is higher than train loss.

4 Time Series Load Forecasting with XGBoost

Xgboost is a gradient boosting library. It provides a laterally boosting trees algorithm that can use to solve machine learning problems. It is available in many languages, for example, C ++, Java, Python, R, Julia, Scala. The project is for getting a small error from all parameters through the Xgboost model in Python. XGBoost is an efficient use of classification gradient intensification and regression problems. XGBoost can also predict time series, although the time series model must first be turned into a guided learning problem.[22]

The basic idea of machine learning model improvement (boosting) is to combine thousands of low-precision predictive models into one high-precision model. With appropriate parameter settings, several models are often required to achieve satisfactory prediction accuracy. If the dataset is large or complex, the model may need to be repeated a lot of times or more to get actual accuracy, and then the XGBoost model can better solve this problem. The XGBoost model was first proposed in 2011 by Chen Tianqi and Carlos Gestrin and has been continuously optimized and refined by many researchers. XGBoost is an efficient and scalable variable for the gradient boosting mechanism.[23]

4.1 Using XGBoost Method For Estonia

In Estonia, XGBoost libraries upload forecasts for each pandemic before analyzing the pandemic actual total load using data and train tests. This chart will help understand how many failures or losses in normal time and Covid-19 were caused by this data analysis through initial test data and expected test data.

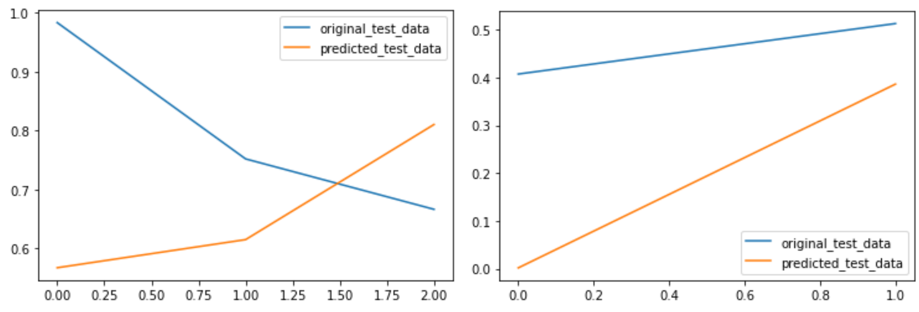

Figure 36. XGBoost Predicted and original prediction for Estonia in before and during lockdown

Experimentally, it shows the difference before lockdown time initial test data and the expected test data. The second diagram shows the same as the original test data and predicted test data. Compared to all these data uses, the behaviour during the value has changes and decreases.

Table 1. XGBoost values in Estonia before lockdown

	STD	RMSE	Loss	MAPE
Hourly train data	0.2098	0.0176	0.01751	2.4694%
Hourly test data	0.1892	0.0565	0.05573	7.0096%
Weekly train data	0.1107	0.0014	0.00138	0.1513%
Weekly test data	0.0416	0.1253	0.13427	15.7472%
Weekend train data	0.2145	0.0047	0.00465	0.5636%
Weekend test data	0.2051	0.2074	0.20597	27.0952%
Monthly train data	0.1695	0.0011	0.00099	0.1127%
Monthly test data	0.1051	0.2871	0.26611	27.3663%

From this table, the XGBoost libraries make a small error in all parameters before lockdown time. Before lockdown, standard deviation, root means square error, Loss function, and means absolute percentage error have changes of LSTM values model. The LSTM model has different result with this same train value which one uses in XGBoost. In this XGBoost have the train STD in hourly 0.2098, where the LSTM model had the hourly STD train value was 0.262. XGBoost libraries use and implement this XGBoost regression in this result have a small error than the LSTM model. Though this table, the RMSE in monthly train data is 0.0011, the LSTM model has the same parameter value was 0.27082. Finally, before the lockdown XGBoost model have an small error than the LSTM model.

Table 2. XGBoost values in Estonia during lockdown

	STD	RMSE	Loss	MAPE
Hourly train data	0.1725	0.0194	0.01877	3.1476%
Hourly test data	0.1385	0.0519	0.05737	inf
Weekly train data	0.2082	0.0009	0.00103	0.2528%
Weekly test data	0.0698	0.1838	0.17901	33.94278%
Weekend train data	0.1761	0.0046	0.00437	0.7364%
Weekend test data	0.1612	0.1932	0.22803	Inf
Monthly train data	0.2203	0.001	0.00126	Inf
Monthly test data	0.1922	0.2859	0.30043	62.1331%

During the lockdown, the uses of energy consumption change dramatically. This table shows a small error value than the LSTM model. IN an LSTM model, monthly test data mean absolute percentage error was 72.45135% which is almost 10% less in the XGBoost model. The loss function also less than the past LSTM model value in train and test data in the XGBoost model. Through table 2, it is understandable how much difference have in LSTM and XGBoost model.

4.2 Using XGBoost Method For Finland

In Finland, load forecasting for any pandemic time and before pandemic time analysis through the train and test data. This chart illustrates how multiple error or loss has been in normal time and in covid-19 time from this data inquiry.

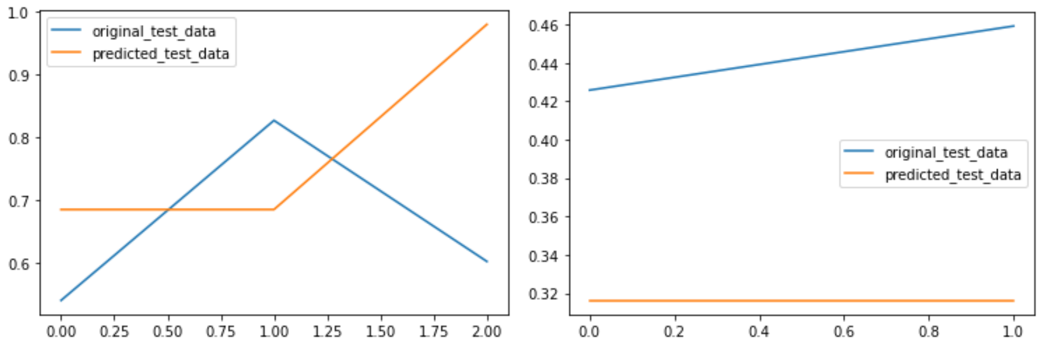

Figure 37. XGBoost Predicted and original prediction for Estonia in before and during lockdown

In observing figure 37 is showing the few difference between before lockdown original test data and predicted test data. Through this analysis, the error is noticeable because it is less than the LSTM model. For comparing all this data, consumption uses behaviour is changes during lockdown time, and it is decreasing.

Table 3. XGBoost values in Finland before Lockdown

	STD	RMSE	Loss	MAPE
Hourly train data	0.1356	0.0065	0.00636	0.7673%
Hourly test data	0.1447	0.0515	0.05151	6.7209%
Weekly train data	0.1113	0.0011	0.00107	0.1293%
Weekly test data	0.0643	0.1581	0.18337	22.119%
Weekend train data	0.1411	0.0028	0.00284	0.4007%
Weekend test data	0.1143	0.1869	0.19373	28.4657%
Monthly train data	0.2129	0.0012	0.00117	0.1615%
Monthly test data	0.1391	0.2406	0.24761	35.5581%

Before lockdown, the user behavior was different than during lockdown. With this XGBoost implementation, the error is small, and for this project, it is better than the previous existing model. This table has all parameters and predicted values. In Finland, before lockdown, the monthly test root means square error was 0.30836, and in this, XGBoost has 0.2406, which is making relevant results in XGBoost. Overall this XGBoost has a small error in train data and test data.

Table 4. XGBoost values in Finland during lockdown

	STD	RMSE	Loss	MAPE
Hourly train data	0.1358	0.007	0.00677	0.8714%
Hourly test data	0.0877	0.0449	0.05664	Inf
Weekly train data	0.1982	0.0008	0.00075	0.1142%
Weekly test data	0.1356	0.3812	0.32782	45.21516%
Weekend train data	0.1432	0.003	0.003	0.4634%
Weekend test data	0.0881	0.1013	0.12638	inf
Monthly train data	0.2478	0.0009	0.00122	inf
Monthly test data	0	0.128	0.12735	28.4333%

During the lockdown, the test predicted error is also smaller than the LSTM model. Implementing this XGBoost, the monthly test means absolute percentage error is 28.4333%, where the LSTM monthly test value was 60.1642%. In this model, the XGBoost decreases the error by more than half. Another value is standard deviation, the monthly test value in LSTM was 0.09320, and in XGBoost, it is fully 0. This is the difference between the normal LSTM and XGBoost models.

4.3 Using XGBoost Method For Norway

Load forecasting for any pandemic and before pandemic time analysis in Norway through the train and test data. During this COVID-19 time and before that time have a different level of user for energy use. Through ENTOs data, the difference can become observable in Norway. The LSTM model and XGBoost give several types of error rates.

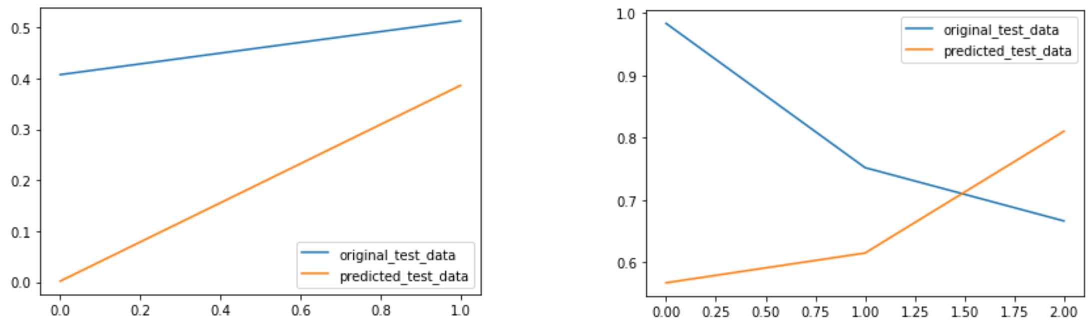

Figure 38. XGBoost Predicted and original prediction for Norway in before and during lockdown

Figure 37 is showing the variation of before lockdown original test data and predicted test data. The second graph is also showing the same thing during lockdown time. By comparing all this data, consumption uses behaviour is changes during lockdown time, and it is decreasing. When the XGBoost values are small, that means the model is suitable for the future pandemic time.

Table 5. XGBoost values in Norway before lockdown

	STD	RMSE	Loss	MAPE
Hourly train data	0.1285	0.0038	0.00375	0.4216%
Hourly test data	0.1225	0.0443	0.04325	4.6238%
Weekly train data	0.1107	0.0014	0.00138	0.1513%
Weekly test data	0.0416	0.1253	0.13427	15.7472%
Weekend train data	0.2145	0.0047	0.00465	0.5636%
Weekend test data	0.2051	0.2074	0.20597	27.0952%
Monthly train data	0.1695	0.0011	0.00099	0.1127%
Monthly test data	0.1051	0.2871	0.26611	27.3663%

Before that pandemic, the past has a different pattern in users. Table 5 is showing the value of XGBoost in all parameters. From this table, the train data and test data are different from the LSTM model. The XGBoost model has a small overall error than the LSTM model. An LSTM model monthly standard deviation test data was 0.1398, and this new XGBoost model has 0.1051. Before lockdown, the monthly test loss function was 4.8784, and in this XGBoost, the same monthly test loss function is 0.26611.

Table 6. XGBoost values in Norway during lockdown

	STD	RMSE	Loss	MAPE
Hourly train data	0.1452	0.0041	0.00402	0.5723%
Hourly test data	0.1006	0.0429	0.0546	Inf
Weekly train data	0.2082	0.0009	0.00103	0.2528%
Weekly test data	0.0698	0.1838	0.17901	33.9427%
Weekend train data	0.1761	0.0046	0.00437	0.7364%
Weekend test data	0.1612	0.1932	0.22803	Inf
Monthly train data	0.2203	0.001	0.00126	Inf
Monthly test data	0.1922	0.2859	0.30043	62.1331%

In Norway, the uses of energy load forecasting in pandemic time have changed in LSTM and XGBoost models. At the time of lockdown, the XGBoost model gives a small amount of error for all parameters. In LSTM, the train loss function was 0.0364, and in XGBoost have 0.00126. So, this variance comes from XGBoost, for this means absolute percentage error also smaller than the previous model.

4.4 Compare Using LSTM and XGBoost Method

In this comparison, have Norway train and test data for understanding the difference between the train and test data in two models. From those tables, the predict result comparison have in two separate way.

In LSTM model has a different value in train data, and XGBoost also has different train data values, which have a huge difference. In all this training, the data value shows the small error in the XGBoost model and will be suitable for prediction. Table 7 shows the value in all parameters and through these values, the XGBoost model is better than the LSTM model for Norway.

Table 7. Train data comparison in LSTM and XGBoost model

Train Data	LSTM before lockdown	XGBoost before lockdown	LSTM during lockdown	XGBoost during lockdown
Hourly Loss, RMSE, STD, MAPE	6.7889 0.0258 0.20613 inf	0.00375 0.0038 0.1285 0.4216	3.9035 0.0258 0.20613 inf	0.00402 0.0041 0.1452 0.5723
Weekly Loss, RMSE, STD, MAPE	0.0761 0.27566 0.21406 inf	0.00138 0.0014 0.1107 0.1513	0.0325 0.27566 0.21406 inf	0.00103 0.0009 0.2082 0.2528
Weekend Loss, RMSE, STD, MAPE	0.0441 0.20957 0.03676	0.00465 0.0047 0.2145	0.011 0.20957 0.03676	0.00437 0.0046 0.1761

	inf	0.5636	inf	0.7364
Monthly Loss, RMSE, STD, MAPE	3.9035 0.27142 0.19685 inf	0.00099 0.0011 0.1695 0.1127	0.0957 0.271424 0.1968 inf	0.00126 0.001 0.2203 inf

From Table 8, the test data also have the difference between in LSTM and XGBoost model. This project has a total of six months of data, and all data is showing the total actual total load. For future prediction, these two models are showing two different types of error. LSTM and XGBoost have different types of errors, but overall the XGBoost has a small error. The calculated test data in both models, the XGBoost, is more suitable for future prediction.

Table 8. Test data comparison in LSTM and XGBoost model

Test Data	LSTM before lockdown	XGBoost before lockdown	LSTM during lockdown	XGBoost during lockdown
Hourly Loss, RMSE, STD, MAPE	0.0011 0.03266 0.20576 70.87186	0.04325 0.0443 0.1225 4.6238	4.8784 0.032662 0.20576 70.87186	0.0546 0.0429 0.1006 inf
Weekly Loss, RMSE, STD, MAPE	0.097 0.306943 0 212.45456	0.13427 0.1253 0.0416 15.7472	0.0698 0.306943 0 213.1092	0.17901 0.1838 0.0698 33.9427
Weekend Loss, RMSE, STD, MAPE	0.0453 0.21278 0.03501 69.76795	0.20597 0.2074 0.2051 27.0952	0.0145 0.21278 0.03501 69.76795	0.22803 0.1932 0.1612 inf

Monthly Loss, RMSE, STD, MAPE	4.8784 0.172715 0.1398 72.623014	0.26611 0.2871 0.1051 27.3663	0.0364 0.172715 0.1398 72.623	0.30043 0.2859 0.1922 62.1331

From table 8, The difference in standard deviation recorded by XGBoost (about 9.22%) shows a significant difference from before and after the pandemic. Therefore, this supports the smaller value (0.2859) of error recorded after the pandemic compared to those recorded before the pandemic(for monthly prediction). During the lockdown, the loss function is 0.00266%, wherein before pandemic time, it was 4.86%(for hourly prediction) error, and comparing the data record, it is suitable. Root means a square error in weekly the percentage error is smaller during the pandemic time. An LSTM model has an RMSE is 0.3069, and XGBoost has 0.1838. From this comparison, we can easily understand that it is a suitable model for future prediction(weekly prediction). The mean absolute percentage error before the lockdown was 72.623, and later through this XGBoost, it was 62.1331. This mean absolute percentage error is 10% less, and it is suitable for new proposed model.

5 Summary

In conclusion, we can understand that energy load forecasting has changeable patterns in pandemic time from the literature review. From this two method we can understand in two way the prediction and XGBoost is better than LSTM model. For various period of load forecasting, we computed performance of various prediction algorithms towards deciding optimal solution for energy forecasting. The combination of LSTM with Boosting method, for an Ensemble model working through XGBoost. The XGBoost performance shows superior suitability for predicting consumption behaviour when compare to error parameter return by LSTM. In contrast to the STD, MAPE, RMSE and Loss function, recorded by LSTM, ensembled model recorded better performance with STD, MAPE, RMSE and Loss function of 0,72.45135%,0.17179,0.0295 respectively(monthly data set) before lockdown. This approch for energy load forecasting works for a short time of prediction during lockdown, based on STD, MAPE, RMSE and Loss function as 0.1922, 62.1331%, 0.0957 , 0.00126 respectively (monthly data set, Estonia). In this research, we are analysing the measurement of future stress on economic activities. This energy load forecast will change the system over a long period during a pandemic. Through the long period of the algorithm, we can manage the load forecast on energy supply, production, and distribution. We are working on maintenance errors and financial damage for the local energy production. This thesis paper builds a system that will maintain the future local power grid and balance to meet the adequate load demand. We are taking input from ENTSO e using the Python notebook tools and the actual budget using the Python library. All this ENTSO e data for predicting stable energy loads. In contrast, the actual management of our local electricity network for long-time load forecasting. This is important to evaluate the distribution or design of the power grid and evaluate the return on investment in the new algorithm and pattern for its expansion.

The estimation of economic actions of future load forecasting in pandemic time based on the two proposed models shows that, XGBoost has a small error which is good for prediction. Through this XGBoost model, it can better manage load forecasting impacts on purchasing, generating, and energy distribution in pandemic time. When we use the XGBoost model for prediction, we easily maintain local energy supply operation parameters during the pandemic and fix financial damage. An LSTM model also good, but in XGBoost have a small error than the LSTM model. Boosting have a sequential method, and the XGBoost is the suitable way that can be made to improve the new method in a long short time in a pandemic.

Based on parameter values, it is sufficient to conclude that among all four timeline(Hourly, Weekly, weekend and Monthly) XGBoost can provide good predictions of the future actual total load. As such, Ensemble outperforms other models to achieve the lowest RMDSE with the trade-off of computing time. In terms of computing time spent on parameter searching and prediction calculation, XGBoost was the fastest one. LSTM works on RMSE and performs calculation while the time taken to tune hyper parameter with respect to XGBoost was usually around a suitable model with less error.

Monthly results before lockdown and during lockdown time in Estonia through LSTM.

Table 9. Before lockdown monthly in LSTM(Estonia)

Parameters	**Train data**	**Test data**
RMSE	0.27082	0.17179
STD	0.19778	0
loss	0.0737	0.0295
MAPE(%)	Inf	72.45135

Table 10. During lockdown monthly in LSTM(Estonia)

Parameters	**Train data**	**Test data**
RMSE	0.27082	0.17179
STD	0.19778	0
Loss	0.0957	0.0383
MAPE	Inf	72.45135

Table 11. Before lockdown monthly in XGBoost (Estonia)

Parameters	**Train data**	**Test data**
RMSE	0.0011	0.2871
STD	0.1695	0.1051
Loss	0.00099	0.26611
MAPE(%)	0.1127	27.3663

Table 12. During lockdown monthly in XGBoost (Estonia)

Parameters	**Train data**	**Test data**
RMSE	0.001	0.2859
STD	0.2203	0.1922
Loss	0.00126	0.30043
MAPE(%)	inf	62.1331

In this paper, we proposed a new model for electricity load forecasting in pandemic time. We converted daily electricity load information into hourly, weekly, weekend, and monthly category. It increases the number of features available for predicting load in.

LSTM model. Then, we used XGBoost, a recently dominant machine learning technique for time series prediction, for feature selection from converted data. Once features are extracted, we train the model using XGBoost. After training, we use a trained model for load prediction.

In the future, work can be done with Adaboost for predicting the energy consumption scenario. Additionally, more effort can be committed to feature extraction to improve the accuracy of the proposed model further.

References

1. M. Czosnyka, B. Wnukowska and K. Karbowa, "Electrical energy consumption and the energy market in Poland during the COVID-19 pandemic," *2020 Progress in Applied Electrical Engineering (PAEE)*, 2020, pp. 1-5, doi: 10.1109/PAEE50669.2020.9158771.

2. Oregon Statewide Eviction Moratorium. (2021). Retrieved 30 April 2021, from https://states.aarp.org/oregon/oregon-statewide-eviction-moratorium

3. J. Faraji, A. Ketabi, H. Hashemi-Dezaki, M. Shafie-Khah and J. P. S. Catalão, "Optimal Day-Ahead Scheduling and Operation of the Prosumer by Considering Corrective Actions Based on Very Short-Term Load Forecasting," in IEEE Access, vol. 8, pp. 83561-83582, 2020, doi: 10.1109/ACCESS.2020.2991482.

4. Abdullayeva, G. (2021). LSTM-arhitektuuride rakendamine ja hindamine energia aegridade prognoosimiseks. Retrieved 15 January 2021, from https://dspace.ut.ee/handle/10062/66421

5. Monitoring of Electrical Consumption, Including Self-Isolation During the COVID-19 Pandemic - IEEE Conference Publication. (2021). Retrieved 16 January 2021, from https://ieeexplore.ieee.org/document/9281179

6. E. R. Norman, “Note from the Editor: COVID-19: Data Reporting, Good Governance, and New Policy Priorities,” *World Affairs*, vol. 183, no. 3, pp. 212–215, 2020.

7. Army FMWR Applied Financial Planning - Forecasting.docx - Module 1 An Overview of Forecasting 1 In simple terms forecasting is equal to a Data | Course Hero. (2021). Retrieved 30 April 2021, from https://www.coursehero.com/file/24437829/Army-FMWR-Applied-Financial-Planning-Forecastingdocx/

8. E. Lisowski. The best forecast techniques or how to predict from time series data, 2019. [Online] Available https://towardsdatascience.com/ the-best-forecast-techniques-or-how-to-predict-from-time-series-data-967221811.

9. E. B.V., “Dimensional Constant,” *Dimensional Constant - an overview | ScienceDirect Topics*. [Online]. Available: https://www.sciencedirect.com/topics/engineering/dimensional-constant. [Accessed: 10-May-2021].

10. N. Yurtoğlu, “http://www.historystudies.net/dergi//birinci-dunya-savasinda-bir-asayis-sorunu-sebinkarahisar-ermeni-isyani20181092a4a8f.pdf,” *History Studies International Journal of History*, vol. 10, no. 7, pp. 241–264, 2018.

11. Nepal, B., Yamaha, M., Yokoe, A., & Yamaji, T. (2021). Electricity load forecasting using clustering and ARIMA model for energy management in buildings. Retrieved 30 April 2021, from

12. Y. Kim and H. Bang, “Introduction to Kalman filter and its applications,” in *Introduction and Implementations of the Kalman Filter*, F. Govaers, Ed. London, England: IntechOpen, 2019.

13. B. Y. S. Paul, “Artificial intelligence vs auditing the future state,” *Theiia.org*. [Online]. Available: https://chapters.theiia.org/atlanta/Presentations/February%202021%20-%20Artificial%20Intelligence%20vs%20auditing.pdf. [Accessed: 03-May-2021].

14. J. Frankenfield, “Artificial Neural Network (ANN),” *Investopedia*, 07-May-2021. [Online]. Available: https://www.investopedia.com/terms/a/artificial-neural-networks-ann.asp. [Accessed: 10-May-2021].

15. M. De Felice and Xin Yao, "Short-Term Load Forecasting with Neural Network Ensembles: A Comparative Study [Application Notes]," in IEEE Computational Intelligence Magazine, vol. 6, no. 3, pp. 47-56, Aug. 2011, doi: 10.1109/MCI.2011.941590.

16. Y. Zhang and Z. Ye, “Short-Term Traffic Flow Forecasting Using Fuzzy Logic System Methods,” *Journal of Intelligent Transportation Systems*, vol. 12, no. 3, pp. 102–112, 2008.

17. Armstrong, J. Scott, Selecting Forecasting Methods (December 16, 2009). Available at SSRN: https://ssrn.com/abstract=1941247 or http://dx.doi.org/10.2139/ssrn.1941247

18. Garg, R., Garg, R., & Garg, R. (2021). A Primer to Ensemble Learning – Bagging and Boosting. Retrieved 30 April 2021, from https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/

19. Many Heads Are Better Than One: The Case For Ensemble Learning - KDnuggets. (2021). Retrieved 30 April 2021, from https://www.kdnuggets.com/2019/09/ensemble-learning.html

20. Brownlee, J. (2021). Ensemble Learning Methods for Deep Learning Neural Networks. Retrieved 30 April 2021, from https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/

21. Brownlee, J. (2021). How to Make Predictions with Long Short-Term Memory Models in Keras. Retrieved 30 April 2021, from https://machinelearningmastery.com/make-predictions-long-short-term-memory-models-keras/

22. Płoński, P. (2021). Xgboost Feature Importance Computed in 3 Ways with Python. Retrieved 30 April 2021, from https://mljar.com/blog/feature-importance-xgboost/

23. Alim, M., Ye, G., Guan, P., Huang, D., Zhou, B., & Wu, W. (2021). Comparison of ARIMA model and XGBoost model for prediction of human brucellosis in mainland China: a time-series study. Retrieved 30 April 2021, from

24. M. Czosnyka, B. Wnukowska and K. Karbowa, "Electrical energy consumption and the energy market in Poland during the COVID-19 pandemic," 2020 Progress in Applied

Electrical Engineering (PAEE), 2020, pp. 1-5, doi: 10.1109/PAEE50669.2020.9158771. https://ieeexplore.ieee.org/document/9158771

25. L. Griffin, "IEEE Xplore. Version 1.3200212Institute for Electrical and Electronics Engineers. IEEE Xplore. Version 1.3. Piscataway, NJ, 2001. < http://www.ieee.org/ieeexplore/> US$59,995-US$91,995 per annum, depending on subscription option", Online Information Review, vol. 26, no. 4, pp. 285-285, 2002. Available: 10.1108/oir.2002.26.4.285.12.

26. H. Zhong, Z. Tan, Y. He, L. Xie and C. Kang, "Implications of COVID-19 for the electricity industry: A comprehensive review," in CSEE Journal of Power and Energy Systems, vol. 6, no. 3, pp. 489-495, Sept. 2020, doi: 10.17775/CSEEJPES.2020.02500.

27. S. Sharma and D. Verma, "Impact of COVID-19 and Nine-Minute Call on Indian Power Sector: Study of Blackout Prevention," *2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP)*, 2020, pp. 301-305, doi: 10.1109/ICT-PEP50916.2020.9249866.

28. D. Thakur, “LSTM and its equations,” *Medium*, 06-Jul-2018. [Online]. Available: https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af. [Accessed: 10-May-2021].

29. E. model, J. Daniel and S. Kolassa, "Ensemble time series model", *Cross Validated*, 2021. [Online]. Available: https://stats.stackexchange.com/questions/47950/ensemble-time-series-model. [Accessed: 05- May- 2021]

30. J. Brownlee, “How to Use StandardScaler and MinMaxScaler Transforms in Python,” *Machine Learning Mastery*, 27-Aug-2020. [Online]. Available: https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/. [Accessed: 08-May-2021].

31. J N, “Elementary Statistics for the rest of us!,” *Statistics How To*, 18-Dec-2020. [Online]. Available: https://www.statisticshowto.com/. [Accessed: 09-May-2021].

32. S. Prabhakaran, “Time Series Analysis in Python - A Comprehensive Guide with Examples,” *ML+*, 08-Sep-2020. [Online]. Available: https://www.machinelearningplus.com/time-series/time-series-analysis-python/. [Accessed: 09-May-2021].

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Mohammad Arif-Uzzaman

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Forecast of the Electricity consumption during pandemic time", supervised by Dr. Juri Belikov and co-supervised by Abiodun Emmanuel Onile.
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

10.05.2021

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 -Description and Steps

This section describes the work and application. It contains a step-by-step how this project get result through LSTM and XGBoost model.

Steps:

1. The folder and files are in this link. In this link have two folder and 1st one is LSTM and the second done is XGBoost https://drive.google.com/drive/folders/15LSiFVHaZdAAEyhifXJvKV3U28nty5Ox?usp=sharing .
2. The program can run on Anaconda Navigator- python Jupyter and colab.
3. For every single python file have single csv file. After import libraries, the second step is import this csv file. Without importing this file, the data will not show in python program file.
4. In every python file have two part, 1st is before lockdown time and the 2nd is during lockdown time.
5. In this project, the compare about pandemic time and before that time error.
6. This link for github user. From github, the code and csv file is available. https://github.com/arif-uzzaman/Thesis-new-project.git

Appendix 3 - Source Code

LSTM

```
#Importing liberaries
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
import math
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings("ignore")



#Loading data
dataset=pd.read_csv('data.csv')
#Estonia country
dataset.head(19)

df1 = dataset[0:2185] # before lockdown
after_df = dataset[2185:] #After lockdown

len(df1)
#Statical data
dataset.describe().transpose()



#Comparing data slices
fig = plt.figure(figsize=(20,8))
plt.subplot(411)
plt.plot(df1, label='Before')
plt.legend(loc='best')
plt.subplot(412)
plt.plot(after_df, label='After')
plt.legend(loc='best')
fig.suptitle('Comparing Before and After Lockdown')
plt.show()

plt.xlabel('Date')
plt.ylabel("Monthly Cases data ")
plt.title(" Compare March , April, May  , January , Feburary , December")
dataset['Actual Total Load [6.1.A]'][0:2185].plot(figsize=[15,5],kind = "line", label =
"Cases Counts January , Feburary , December")
```

```
dataset['Actual Total Load [6.1.A]'][2185:].plot(figsize=[15,5],kind = "line" ,color =
"green",label = "Cases Counts March , April, May " )
plt.legend(loc='best')
plt.show()

#Plotting the graph
plt.xlabel('Date')
plt.ylabel("Monthly Cases data ")
plt.title("data Before lockdown  ")
df1['Actual Total Load [6.1.A]'].plot(figsize=[15,5], label = "Cases Counts")
plt.legend(loc='best')

df1

scaler=MinMaxScaler(feature_range=(0,1))
df1=scaler.fit_transform(np.array(df1['Actual Total Load [6.1.A]']).reshape(-1,1))
print(df1)

##splitting dataset into train and test split
training_size=int(len(df1)*0.7)
test_size=len(df1)-training_size
train_data,test_data=df1[0:training_size,:],df1[training_size:len(df1),:1]

training_size,test_size

import numpy
# convert an array of values into a dataset matrix
def create_dataset(dataset, time_step=1):
	dataX, dataY = [], []
	for i in range(len(dataset)-time_step-1):
		a = dataset[i:(i+time_step), 0]   ###i=0, 0,1,2,3-----99   100
		dataX.append(a)
		dataY.append(dataset[i + time_step, 0])
	return numpy.array(dataX), numpy.array(dataY)

# reshape into X=t,t+1,t+2,t+3 and Y=t+4
time_step = 41
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

print(X_train.shape), print(y_train.shape)
print(X_test.shape), print(y_test.shape)

# reshape input to be [samples, time steps, features] which is required for LSTM
X_train =X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
X_test = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)

### Create the Stacked LSTM model
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(41,1)))
```

```
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=81,batch_size=64,ver
bose=2)

### Lets Do the prediction and check performance metrics
ori_train_predict=model.predict(X_train)
ori_test_predict=model.predict(X_test)

##Transformback to original form
train_predict=scaler.inverse_transform(ori_train_predict)
test_predict=scaler.inverse_transform(ori_test_predict)
```

RMSE, STD, MAPE

```
def mean_absolute_percentage_error(y_true, y_pred):
   y_true, y_pred = np.array(y_true), np.array(y_pred)
   return np.mean(np.abs((y_true - y_pred) / y_true), dtype=np.float64) * 100

### Calculate RMSE performance metrics
train_RMSE_before_lockdown = math.sqrt(mean_squared_error(y_train,
ori_train_predict))
print('train_RMSE_before_lockdown :', train_RMSE_before_lockdown)

train_STD_before_lockdown = ori_train_predict.std(axis = 0)
print('train_STD_before_lockdown :', train_STD_before_lockdown)

train_MAPE_before_lockdown = mean_absolute_percentage_error(y_train,
ori_train_predict)
print('train_MAPE_before_lockdown :', train_MAPE_before_lockdown)

### Test Data RMSE
test_RMSE_before_lockdown = math.sqrt(mean_squared_error(y_test,
ori_test_predict))
print('test_RMSE_before_lockdown', test_RMSE_before_lockdown)

test_STD_before_lockdown = ori_test_predict.std(axis = 0)
print('test_STD_before_lockdown :', test_STD_before_lockdown)

test_MAPE_before_lockdown = mean_absolute_percentage_error(y_test,
ori_test_predict)
print('test_MAPE_before_lockdown :', test_MAPE_before_lockdown)
plt.plot(y_train)
plt.plot(ori_train_predict)
plt.legend(["original_train_data", "predicted_train_data"])
plt.show()

plt.plot(y_test)
plt.plot(ori_test_predict)
```

```
plt.legend(["original_test_data", "predicted_test_data"])
plt.show()

### Plotting
# shift train predictions for plotting
plt.figure(figsize=[15,5])
plt.xlabel('Observations ')
plt.ylabel("Monthly Cases data ")
plt.title(" Actuals and predictions  ")
look_back=41
trainPredictPlot = numpy.empty_like(df1)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict)+look_back, :] = train_predict

# shift test predictions for plotting
testPredictPlot = numpy.empty_like(df1)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:len(df1)-1, :] = test_predict

# plot baseline and predictions
plt.plot(scaler.inverse_transform(df1), label = "Original data")
plt.plot(trainPredictPlot , label = "Training data")
plt.plot(testPredictPlot , color = "red" , label = "Testingn data")
plt.show()
```

XGBoost

```
#import libraries
import numpy as np
import pandas as pd
from sklearn.metrics import mean_absolute_error
from matplotlib import pyplot as plt
from xgboost import XGBRegressor
from sklearn.model_selection import cross_val_score
from sklearn.metrics import median_absolute_error, mean_absolute_error
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.preprocessing import MinMaxScaler
import warnings
warnings.filterwarnings("ignore")

# load the dataset
df = pd.read_csv('data.csv')
scaler=MinMaxScaler(feature_range=(0,1))
df=pd.DataFrame(scaler.fit_transform(np.array(df)))
print(df.head())
before_lockdown = df[0:2185]
during_lockdown = df[2185:]
print(df.shape)
print(before_lockdown.shape)
print(during_lockdown.shape)
```

```
#xgboot
model = XGBRegressor(n_estimators=20000, objective ='reg:squarederror',
booster='gbtree', colsample_bytree = 0.3,
                    learning_rate = 0.01, max_depth = 5)

# Cross Validation
cv_results = cross_val_score(model, X_train, Y_train, cv = 5, scoring='r2', n_jobs = -1,
verbose = 1)
prt_string = "CV Mean R2 score: %f (Std: %f)"% (cv_results.mean(), cv_results.std())
print(prt_string)

#Train data
pred_Value        = trained_Model.predict(X_train)
r2_val           = r2_score(Y_train, pred_Value)
m_err_val         = median_absolute_error(Y_train, pred_Value)
mean_err_val       = mean_absolute_error(Y_train, pred_Value,
                                 sample_weight = Y_train, multioutput='uniform_average')
mean_sqr_err_Value   = mean_squared_error(Y_train, pred_Value, sample_weight =
Y_train, multioutput='uniform_average')
rmse_Value          = np.sqrt(mean_sqr_err_Value)
yMax_Value          = np.max(Y_train)
yMin_Value          = np.min(Y_train)
nrmse_Value         = rmse_Value / (yMax_Value - yMin_Value)
MAPE              = mean_absolute_percentage_error(Y_train, pred_Value)
STD              = pred_Value.std(axis = 0)

#test
pred_Value        = trained_Model.predict(X_test)
r2_val           = r2_score(Y_test, pred_Value)
m_err_val         = median_absolute_error(Y_test, pred_Value)
mean_err_val       = mean_absolute_error(Y_test, pred_Value,
                                 sample_weight = Y_test, multioutput='uniform_average')
mean_sqr_err_Value   = mean_squared_error(Y_test, pred_Value, sample_weight =
Y_test, multioutput='uniform_average')
rmse_Value          = np.sqrt(mean_sqr_err_Value)
yMax_Value          = np.max(Y_test)
yMin_Value          = np.min(Y_test)
nrmse_Value         = rmse_Value / (yMax_Value - yMin_Value)
MAPE              = mean_absolute_percentage_error(Y_test, pred_Value)
STD              = pred_Value.std(axis = 0)
```

Table 13. Train data comparison in Estonia

Train **Data**	**LSTM before lockdown**	**XGBoost before lockdown**	**LSTM during lockdown**	**XGBoost during lockdown**
Hourly Loss, RMSE, STD, MAPE	7.1553 0.0286 0.262 inf	0.01751 0.0176 0.2098 2.4694	5.8068 0.02861 0.26202 inf	0.01877 0.0194 0.1725 3.1476
Weekly Loss, RMSE, STD, MAPE	0.0386 0.1965 0.02382 40.70348	0.00138 0.0014 0.1107 0.1513	0.0386 0.1965 0.02382 40.70348	0.00103 0.0009 0.2082 0.2528
Weekend Loss, RMSE, STD, MAPE	0.0625 0.24963 0.03566 inf	0.00465 0.0047 0.2145 0.5636	0.0625 0.24963 0.03566 inf	0.00437 0.0046 0.1761 0.7364
Monthly Loss, RMSE, STD, MAPE	0.0737 0.27082 0.19778 inf	0.00099 0.0011 0.1695 0.1127	0.0737 0.27082 0.19778 inf	0.00126 0.001 0.2203 inf

Table 14. Test data comparison in Estonia

Test Data	**LSTM before lockdown**	**XGBoost before lockdown**	**LSTM during lockdown**	**XGBoost during lockdown**
Hourly Loss, RMSE, STD, MAPE	0.0013 0.0354 0.2345 67.5417	0.05573 0.0565 0.1892	6.6634 0.03407 0.24	0.05737 0.0519

		7.0096	66.36274	0.1385 inf
Weekly Loss, RMSE, STD, MAPE	0.0541 0.2326 0.0151 64.5397	0.13427 0.1253 0.0416 15.7472	0.1616 0.2326 0.0151 64.53977	0.17901 0.1838 0.0698 33.94278
Weekend Loss, RMSE, STD, MAPE	0.0546 0.23376 0.02096 51.62104	0.20597 0.2074 0.2051 27.0952	0.0294 0.23376 0.02096 51.62104	0.22803 0.1932 0.1612 inf
Monthly Loss, RMSE, STD, MAPE	0.0295 0.17179 0 72.45135	0.26611 0.2871 0.1051 27.3663	0.0383 0.17179 0 72.45135	0.30043 0.2859 0.1922 62.1331

This two tables shows the difference between pandemic time and during the pandemic time . This two model have two difference error and it's show the small error in XGBoost model. So, in ensemble the XGBoost model is better than LSTM model.

Table 15. Train data comparison in Finland

Train Data	LSTM before lockdown	XGBoost before lockdown	LSTM during lockdown	XGBoost during lockdown
Hourly Loss, RMSE, STD, MAPE	8.337 0.027982 0.17668 inf	0.00636 0.0065 0.1356 0.7673	6.8858 0.0258 0.17668 inf	0.00677 0.007 0.1358 0.8714
Weekly Loss, RMSE, STD, MAPE	0.0181 0.14697 0.02184 30.78956	0.00107 0.0011 0.1113 0.1293	0.0592 0.14697 0.02184 30.78956	0.00075 0.0008 0.1982 0.1142
Weekend Loss, RMSE, STD, MAPE	0.0298 0.17249 0.06576 58.6831	0.00284 0.0028 0.1411 0.4007	0.0116 0.172498 0.06576 58.683104	0.003 0.003 0.1432 0.4634
Monthly Loss, RMSE, STD, MAPE	0.0803 0.283906 0.1295 132.15419	0.00117 0.0012 0.2129 0.1615	0.0948 0.283906 0.12951 132.15419	0.00122 0.0009 0.2478 inf

Table 16. Test data comparison in Finland

Test Data	LSTM before lockdown	XGBoost before lockdown	LSTM during lockdown	XGBoost during lockdown

Hourly Loss, RMSE, STD, MAPE	0.0014 0.037508 0.19063 80.8097	0.05151 0.0515 0.1447 6.7209	0.0011 0.032662 0.19063 80.8097	0.05664 0.0449 0.0877 inf
Weekly Loss, RMSE, STD, MAPE	0.1165 0.2326 0.0151 64.53977	0.18337 0.1581 0.0643 22.119	0.1445 0.295209 0.007837 57.45321	0.32782 0.3812 0.1356 45.21516
Weekend Loss, RMSE, STD, MAPE	0.0308 0.175449 0.06738 28.97608	0.19373 0.1869 0.1143 28.4657	0.015 0.175449 0.06738 28.97608	0.12638 0.1013 0.0881 inf
Monthly Loss, RMSE, STD, MAPE	0.0969 0.3083619 0.093209 60.16426	0.24761 0.2406 0.1391 35.5581	2.5469 0.30836 0.093209 60.16426	0.12735 0.128 0 28.4333

In Finland the XGBoost model error is more smaller than the LSTM model. So, in the