Arab American University
Faculty of Graduate Studies

Electrical Load Forecasting Based on Machine Learning Approach for Palestine's Power System

By

Mobarak Abdulmutie Hassan Abu Mohsen

Supervisor

Dr. Amani Owda

Co-Supervisor

Dr. Majdi Owda

This thesis was submitted in partial fulfillment of the requirements for the Master's degree in data science and business analytics

February / 2023

Thesis Approval

Electrical Load Forecasting Based on Machine Learning Approach for Palestine's Power System

By

Mobarak Abdulmutie Hassan Abu Mohsen

This thesis was defended successfully on 12- February -2023 and approved by:

Committee members	Signature
1. Dr. Amani Owda: Supervisor	
2. Dr. Majdi Owda: Co-Supervisor	
3. Dr. Huthaifa Ashqar: Internal Examiner	
4. Dr. Radi Jarrar: External Examiner	

Declaration

I declare that the thesis titled "Electrical Load Forecasting based on Machine Learning Approach for Palestine's Power System" is my work, has been composed solely by myself and does not contain work from other researchers, and has not been submitted for any other degree or scientific work except the reference is made.

Name: Mobarak Abdulmutie Hassan Abu Mohsen

Student ID: 202012194

Signature:

Date: 17/September / 2024

Dedication

I dedicate this thesis to my family and friends for their unconditional love and support. To my mother and father, for their support, which has not left me throughout my life. Also for my brothers and sisters, whose support I have not always forgotten. My dear friends and work colleagues, for their continued support throughout my learning and work journey.

Acknowledgment

I would like to use this space to express my deep gratitude to Dr. Amani Owda and Dr. Majdi Owda for their advice, help, and valuable time; they spent reviewing and correcting my work. Dr. Amani & Dr. Majdi provided helpful suggestions and advice that have had an important effect and helped in overcoming many obstacles in preparing this work in the best way possible.

Abstract

Forecasting the future electrical load requirements is the first step in power system design and growth. Forecasting the electrical load is critical from both a technical and a financial standpoint. It is essential in traditional commercial procedures and electricity transmission planning to improve power system performance, reliability, safety, and stability while also lowering operating costs. Because of its relevance in power management, infrastructure planning, and budgetary, electricity load forecasting has piqued the interest of researchers and businesses.

The actual electrical load data was obtained from the Electricity Distribution Company in Palestine - Tubas District Electricity Company for about a year. This data is based on the electrical loads taken by the Electricity Company from the connection point with the Israeli side, for the total demand for loads every minute. The main objective of this work is to make a forecasting model and accurate calculation of electrical load based on the measurements of the current loads in Palestine. The importance of having model for forecasting electrical loads will reduce the cost and resources, especially in Palestine because there are no research studies conducted to forecast these loads.

Deep learning is used to forecasting the electrical loads using long short-term memory (LSTM), gated recurrent units (GRU), and recurrent neural networks (RNN). All these models are tested to obtain the best results and the lowest error rate compared to the previous studies. The tuning parameters were adjusted based on several methods, namely: learning rate, type of optimizers, training and testing ratio, activation functions, batch size, epoch, and the number of hidden layers. The models are tested with different approaches in tuning the parameter to achieve the best performance. The GRU model obtained the best model in terms of accuracy and the lowest

error rate when applying two hidden layers and an Adam optimizer, with a test rate of 70% and a number of epochs = 50, and a batch size is 32. The results showed that the GRU model achieved an R^2 of 90.228%, MSE = 0.00215, and MAE = 0.03266.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ML	Machine Learning
AI	Artificial Intelligent
EMD	Empirical Mode Decomposition
STLF	Short-Term Load Forecast
MTLF	Mid-Term Load Forecast
LTLF	Long-Term Load Forecast
NN	Neural Networks
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Networks
GRU	Gated Recurrent Unit
MAE	Mean Absolute Error
MSE	Mean Square Error
RMSE	Root Mean Square Error
R^2	Coefficient Of Determination
SCADA	Supervisory Control And Data Acquisition
TDECO	Tubas District Electricity Company
Adagrad	Adaptive Gradient Descent
RMSProp	Root Mean Square Propagation
GA	Genetic Algorithm
VMD	Variation Mode Decomposition
BOA	Bayesian Optimization Algorithm
AEP	American Electric Power
TD-CNN	Time-Dependency Convolutional Neural Network

C-LSTM	Cycle-Based Long Short-Term Memory
ELM	Extreme Learning Machine
BPNN	Back-Propagation Neural Network
KNN	K-Nearest Neighbor Regression
RF	Random Forest
RT	Regression Tree
SVR	Support Vector Regression
NARX	Non-Linear Autoregressive Exogenous
CNN	Convolutional Neural Networks
ANFIS	Adaptive Neuro-Fuzzy Inference System
SC	Subtractive Clustering
FCM	Fuzzy C Means
GP	Grid Partitioning
ARIMAX	Auto-Regressive Integrated Moving Average With Exogenous
RBFNN	Radial Basis Function Neural Networks
FFNN	Feedforward Neural Networks
SGD	Stochastic Gradient Descent

Chapter One

Introduction

1.1 Electrical Load Forecasting Introduction

The last era in the world was generally characterized by the rapid and large expansion of electricity networks, especially electrical loads, as they swelled dramatically and new types of these electrical loads appeared that need a special study [1]. The increase in electrical loads also made it difficult to design the components of the electrical system. The reorganization of the energy system also led to the formation of institutionalized generation, transmission and distribution companies. These entities are challenged by the increasing requirements for the reliable operation of power system networks [2]. The main concern of every electrical company is to provide reliable and continuous service to its customers.

It has become difficult to forecast electrical loads using traditional and old methods since many factors affect electrical loads directly and indirectly. Those factors are population census, temperatures, climatic changes, rainwater, and underground basins, the economic system in each country, human behavior, global epidemics, and the evolution of industries [3].

Electricity in Palestine is taken from the Israeli occupation through connection points between the two sides, and some of these points have high electricity consumption and others low consumption, which causes malfunctions in high-load transformers and leads to problems in energy outputs and infrastructure. Electrical load forecasting is critical in establishing and improving power system efficiency because it ensures reliable and economical planning, control, and operation of the power system. It helps the electricity companies to make critical choices such as the acquisition and generation of electrical power, as well as the establishment of the infrastructure for the transmission and distribution system.

1.2 Problem Statement

With the rapid and dramatic increase in energy consumption, developing reliable models to forecasting electrical loads is becoming increasingly difficult [4,5]. The problem of rapid and sharp growth in energy consumption in Palestine has led to the need to create reliable models to forecasting electricity loads, as these models will help the electricity companies in managing and planning energy transmission and ensuring reliable and uninterrupted service to their customers. Forecasting electrical loads is very important for electricity companies in Palestine to prepare plans, and it is also important for the energy authority and government agencies to secure energy in the coming years.

Forecasting electrical loads do not depend only on the power sector in Palestine, but instead requires the concerted efforts of all the steps and economic sectors in the country, as all of them must feed this type of study. Studies of the development of loads can feed all economic sectors for feedback that may benefit them in preparing plans for future development for these sectors.

Globally, the importance of forecasting loads comes from the difficulty of forecasting loads are the missing and most ambiguous link for many countries that seek to develop strategies for the electrical system in general. Load forecasting is not only useful in the design of electrical networks, but also to develop strategic plans that ensure a stronger economy, a cleaner environment, and energy sustainability.

1.3 Objectives

In this research, one of the main objectives is to forecast short-term electrical loads in Palestine based on real historical data and regression algorithms (LSTM, GRU, and RNN). The objectives of this work can be illustrated in those points:

- Forecasting electricity loads with the highest accuracy simulates the real development of electrical loads.

- Assisting electrical companies in developing short and medium-term plans in designing electrical networks and estimating infrastructure needs.
- Improving the electricity service in Palestine and solving the problem of power outages in Palestine.
- Helping the electricity companies in securing sources of energy that are suitable for the loads and not reduce the loads, as this increase is considered a waste that cannot be used.
- Using Python layout libraries to visualize data sets to get a deep insight into the data will help us understand the data set and the correlation between the elements.
- Compare the results while using different deep learning algorithms.

1.4 Contribution

The main contribution of this study is to build a machine-learning model to forecast load electricity based on real data in Palestine, this model will help the electricity companies in Palestine to introduce reliable and uninterrupted services to their customers. Moreover, the proposed model will help the electric companies to make critical decisions, such as the development of transmission and distribution systems infrastructure to guarantee the best electrical services for the customers. These models were also built based on several features available such as (day, month, temperature, and electrical load). To the best of our author's knowledge, there are no studies in the open literature conducted to forecast the electrical loads in Palestine using real data in machine-learning algorithms.

1.5 Thesis Structure and Summary

The rest of the thesis is organized as follows. Chapter 2 represented the literature review of the relevant works related to the forecast of short-term electrical loads, the forecast , and loads of other periods electrical loads, the forecast of household electrical loads, and the algorithms they relied on in forecasting. In Chapter 3, exploratory data analysis will be performed to show the data set graphically and items to gain deeper insights from the data set. In Chapter 4, the proposed method for regression using three machine learning algorithms for each task and methods for obtaining and fine-tuning the best parameters for use in deep learning models will be demonstrated. Chapter 5, the results of regression using various rating scales will be presented and discussed. Finally, Chapter 6 will represent the conclusion and future work.

Chapter 2

2. Literature Review

This chapter presents the state of the art and analysis of the relevant literature review for the forecasting of electrical loads and demands. Moreover, different approaches of machine learning algorithms used for short-term, long historical data electrical load forecasting are presented and discussed.

2.1 Background

Machine learning(ML) and deep learning algorithms are widely used in the field of forecasting energy demand and the amount of electricity consumption [6]. Engineers and data scientists depend on these approaches to deal with temporal data in terms of exploration, explanation, and analysis. Two broad categories may be identified in the literature ML application are: 1) unsupervised learning techniques, which are primarily used to give descriptive analytics or as pre-processing stages and discover the behavior of electricity consumption [7, 8, 9, 10], and 2) supervised learning approaches, which are mostly used for predictive modeling [11, 12, 13, 14, 15].

One of the most obstacles that face the Palestinian electricity companies is forecasting the electricity loads since the forecasting process helps these companies guarantee the electrical services to their customers, reduce power outages, and management of their electrical network, so there is a need for building a reliable electricity load forecasting system to forecasting future power loads in Palestine. It is worth mentioning not possible to depend on the power loads Forecasting system that had been proposed in the literature by the previous researchers since the weather factors and the power sources differ from country to country.

The following sections of this chapter are structured as follows: Section 2.2 presents electrical load forecasting; Section 2.3 views studies related to short-term electrical load forecasting; Section 2.4 displays the literature related to long historical data electrical load forecasting; and section 2.5 present conclusions from previous research and studies.

2.2 Electrical Load Forecasting

Load forecasting can be classified as engineering techniques or information methodologies techniques. Engineering techniques, often defined as physics-based modeling, employ physical rules to forecast and evaluate power consumption. To compute energy use, these models frequently rely on contextual elements such as the structure of the building, meteorology, heated, ventilated, and heating and cooling information. Where you need many mathematical operations and many equations to try to forecasting, knowing that the accuracy is low, and it takes a lot of time to compensate for the variables and find the value of each variable of the variables. Building energy simulator tools such as energy plus and equest use physics-based models [16]. The limitations of these models originate from their reliance on the accessibility and accuracy of the training dataset [17].

Data-driven techniques, often known as artificial intelligence (AI) methodologies, are based on past energy consumption data. Many research groups are interested in these techniques, although little is known regarding their forecasting generality. In the other words, when presented with new electricity usage, their precision drops. Data-driven techniques were used in [18] to solve the problem of instability, non-linearity, and seasonal multiplicity because data-driven techniques extract relevant and meaningful information from the available data. In [19], [20] researchers used deep learning optimization techniques for short term load forecasting (STLF) and data-driven study of dynamical models (dynamic mode deconstruction).

The ability of a data-driven forecasting model to generalize remains a key problem as forecasting outcomes vary dramatically for micro-grids with various capacities and load characteristics. A model in [21] was built to forecasted short-term loads for small networks, Using Particle Swarm Optimization (PSO), Empirical Mode Decomposition (EMD), and Extreme Learning Machine with Kernel (KELM). EMD was used for time-series data analysis, PSO for optimizing parameters in the model, and KELM for forecasting after data processing. The results showed that RMSE = 16.081 in the first week and 18.3351 in the fourth week, and this indicates that the algorithm that was used needs to improve the parameters.

The availability of a large amount of energy-related information makes it increasingly desirable to apply data-driven strategies for load forecasting over multiple timeframes rather than physics-based techniques (short, intermediate, and long historical data). Load forecasting may be roughly categorized into three major types based on time scale [22]:

- Short-term load forecast (STLF): The duration of STLF might be a few minutes, hours, days ahead, and weeks. STLF attempts for cost-effective dispatching and optimum generating dispatchable while simultaneously addressing actual safety and control evaluation, which frequently necessitates prior load and weather data.
- Mid-term load forecast (MTLF): MTLF has a period ranging from a month to a year or two. MTLF's purpose is to terms of purchasing power and supply through equipment maintenance, load dispatch coordination, and pricing settlement. MTLF requires meteorological and economic data.
- Long-term load forecast (LTLF): The time period of LTLF is more than one year to 20 years. LTLF attempts to plan system growth, including generation, transmission, and distribution. It also has an impact on the acquisition of new generating units in some situations and requires meteorological, economic, demographic, and land use data.

Because of its utility in management control, energy usage, power storage operations, peak usage anticipation, energy shortage reduced risk, and other applications, short-term load forecasting (STLF) has increased in popularity in buildings, smart grids, and micro - grids [23].

There are two sorts of forecasting techniques: quantitative and qualitative [24], [25]. Quantitative approaches include moving averages [26], regression methods [27], time series [28], exponential smoothing [29], and trend projection [30]. They are employed when the situation is steady and previous data is available. It also employs mathematical approaches. whereas qualitative procedures [31], such as Delphi methodologies [32] and high-level experts [33], are employed. They are employed when the situation is ambiguous and there is limited available data. They involve intuition and, at times, statistical models to supplement it.

2.3 Short-Term Load Forecasting (STLF)

STLF is essential to lower the cost of electricity transactions, encourage the reliable functioning of smart grids without interruption, and control loads as needed. STLF is also used to assess the risks of electricity shortages and reduce the appearance of loads to obtain a stable and reliable electrical network. This section will talk about the following subsections: short-term load forecasting for medium and large electrical networks, and short-term load forecasting for small electrical networks.

2.3.1 Short-Term Load Forecasting For Medium and Large Electrical Networks

The models were built [34, 35, 36] using Neural Networks (NN) since NN can be explained by the network's capacity to learn complicated correlations between input and output patterns that would be impossible to represent using traditional approaches. Inputs to the networks are often current and historical load values. The network is trained using historical real-load data. In addition, based on real data, they used different methods to reach the best result using different optimizers. In [34] used and applied the new model in Bangladesh for electricity demand management, capacity planning, and electric grid increase, to forecast the load within the next few hours to a few weeks, where they used integration of convolutional neural network (CNN) and long short-term memory (LSTM) network. In addition to, other ML models such as XGBoost, and RBFN. The result shows that LSTM-CNN achieved high accuracy compared with other models; Table 2.1 shows the results of using CNN with LSTM was less MAE, less RMSE, and less MAPE compared with other algorithms. [35] Using and applying the new model in Jordan, optimal operational approaches, like genetic algorithm (GA), elephant herding optimization (EHO), and particle swarm optimization (PSO), were used to improve the upgraded parameters, and the authors estimated the errors before and after the improvement. In this research [35], the percentage error (PE) was computed as described in Equation (1), ME, the MSE, and RMSE. The primary takeaway from the research for the authors in [35] is that a two-layer NN is the optimum for load forecasting, Table 2.2 illustrates the results. [36] Proposed a method based on neural networks and particle swarm optimization (PSO) to evaluate the Iranian power system. The neural network-based solutions resulted in fewer forecasting mistakes due to their capacity to adapt effectively to the hidden properties of the consuming load, and neural network parameters have a substantial impact on its performance. The PSO method is used in certain NN to properly anticipate electrical load by defining characteristics such as learning rate and hidden layer count. Using these updated parameters, a three-layer FFNN trained by a backpropagation method and an enhanced PSO method is then used to forecasting short-term electrical demand. The result of the proposed method MAPE is 0.0338 and MAE is 0.02191.

Table 2.1: MAE, RMSE, MAPE for LSTM, CNN-LSTM, RBFN and XGBOOST [34].

MAE				RMSE				MAPE%			
LSTM	CNN_LSTM	RBFN	XGBOOST	LSTM	CNN_LSTM	RBFN	XGBOOST	LSTM	CNN_LSTM	RBFN	XGBOOST
486.5	304.8	558.49	664.5	557.9	429.8	615.2	755.45	5.71	4.23	6.23	7.98

Table 2.2: FORECASTED ERROR FOR THE SYSTEM[35].

Method	PE %	MSE	RMSE	ME	Error St-D
NN- double layers	0.3769	10272.729	101.355	6.0406	101.4768
NN-PSO-single layer	2.4301	68167.516	261.0891	19.7456	264.119
NN-GA- single layer	10.4183	174354.85	417.558	185.97	374.977
NN-EHO- single layer	19.84	290998.21	539.4425	419.904	339.655

$$PE = \frac{|actual\ load - forecasted\ load|}{actual\ load} \qquad (1)$$

A short-term electrical load forecasting system in [37] was used to forecasting the load in the coming hours in North America and New England, based on real data and machine learning algorithms (using CNN, LSTM), initially, they extracted the features of the load based on CNN, gated LSTM-GRU was applied for its ability to retain short and long term memories. The multi-dimensional features recovered by 2-d CNN are then provided as input to the unidirectional propagation GRU and LSTM unit to conduct hourly load power forecasting, with the results shown by arranging the electrical load time - series data and temperatures time - series data in two-dimensional frames. The achieved results are illustrated in Table 2.3.

Table 2.3: Performance of the proposed model in [37].

RMSE (MW)	MAE (MW)	MAPE (%)
80.20	66.12	0.37

Also, models in [38, 39, 40, 41] were proposed based on convolutional neural network algorithms, but each of the researchers used different methods to get the best result for short-term load forecasting. In [38] model was built in China, relying on historical real data and machine learning algorithms, the forecasting model was trained by loading the 7-day historical feature and 21 days, 99 days, and 199 days using inverse neural network, packing regression, and multi-spatial time scale temporal convolutional network. They first applied a multi-temporal-spatial-scale (MTSS) approach using statistical methods to clean up the data samples and enhance their time series features. After that, they used a multi-temporal convolutional network (MTCN), and the MTCN dilated causal convolution layer which has retention of data and the ability to pick up on time series characteristics. This allows the MTCN to anticipate short-term load more correctly, where it was shown that a CN trained on data from many geographic scales could forecast short-term power loads with greater precision, $R^2 = 0.96$, and mean absolute percentage error=1.8. In [39] a model based on CNN evolved in which the statistical properties of each time series data set were exploited to optimize the neural network's hyper-parameters. In addition to transforming the given data set into a model that allows using the CNN algorithm's maximum benefits, the suggested algorithm was compared to the LSTM approach. Moreover, the experimental findings demonstrated solid evidence for the usefulness of the proposed methodology as shown in Table 2.4. Researchers in [40] built a model to forecast load requirements in Jordan Relying on real historical data for one year containing total daily demand per hour and machine learning algorithms (neural network) using the last d feed-forward gray wolf (FFGWO) optimizer. The results showed that by using this optimizer (GWO), the best prediction accuracy and less error were obtained, where MAE = 221.28, and RMSE = 277.91, but Logistic regression gave MAE = 271.07, and RMSE = 342.27. Using a CNN model and a data-augmentation strategy that may artificially expand the training sets, researchers have shown how to overcome difficulties brought on by a lack of historical data and improve the precision of home load forecasting [41]. Due to an increase in the mistake rate caused by a lack of previous data. As a result, when applied to augmentation of the data, the MAPE 19.26%, and RMSE 0.1666 were less compared without augmentation; however, one of the disadvantages of this method is that it can lead to bias.

Table 2.4: Summary of results in [39].

Models	LSTM	CNN	MLP	ANN
MAE	4.58	3.93	5.27	7.58
RMSE	5.3	4.67	4.56	6.63

The studies in [42], [43] used empirical modal decomposition (EMD), the raw data on power use is first divided into a number of intrinsic mode functions (IMFS) of varying frequencies and amplitudes. Researchers in [42] gated recurrent units with feature selection as indicated by empirical mode decomposition for short-term load forecasting. Using empirical mode decomposition, break down the original series into subseries, and then utilize the Pearson correlation as an input feature into the prediction model to find out how closely the subseries are related to the original series, using both the original series and the high-correlation sub-series to train the GRU network. The experiment findings revealed that the suggested method's average prediction accuracy on four data sets was 96.9 %, 95.31 %, 95.72 %, and 97.17 %, consecutively. Moreover, Improved short-term load power consumption forecasting was given in [43] using a combined EMD and LSTM. First, EMD decomposes the raw data on power usage into intrinsic mode functions (IMFS) of varying frequencies and amplitudes. After that, applying LSTM, an intelligent machine learning technique, to extract characteristics and produce temporal predictions for each individual component of the IMF. Finally, by combining the predictions of several targets, data for short-term end-user power usage may be obtained. The proposed EMD - LSTM method achieved MAPE= 2.6249 % in the winter and 2.3047 % in the summer.

Researchers in [44] built a model to accurately forecast short and medium electrical loads, and more than one model was developed to compare and select accurate time series models based on basic feature engineering and to find the optimal periods. It relied on machine learning based on LSTM, and the best features were selected using envelope selection and inline feature methods. In addition, they used a GA to better predict future outcomes of the LSTM model based on the data about power consumption in France, it is necessary to choose the ideal time intervals as well as the number of layers. The results showed that the LSTM -based model Demonstrates high accuracy and the prediction accuracy has been improved by optimization of LSTM features, delays, layers, and training across many LSTM architectures, and optimized with hyperparameter tuning, Table 2.5 shows the result of this model.

Moreover, in China, a hybrid short-load forecasting system based on VMD and LSTM networks and optimized using the BOA has been developed [45]. They compared the proposed methods with SVR, multi-layered perceptron regressor, LR, RF, and EMD-LSTM, the result of the proposed method shows that MAPE is 0.4186% and R2 is 0.9945. In addition, a short-term electrical load forecasting model was built [46] because of the crucial role, it plays in ensuring the reliability, longevity, and efficiency of the power generation and schedule process. With precise load forecasting, power system managers may make confident choices. For precise short-term power load forecasting using a GRU and actual data from AEP, the model in [46] relied on a decoder architecture-based encoder-decoder network with Bayesian optimization. Which showed that using deep learning gave reliable and reliable forecasting accuracy in providing a time-series electrical load forecasting relay strategy. Table 2.6 shows the result of the proposed method.

Table 2.5: Analysis of LSTM-Time-Horizon-Dependent RNN's Performance [44].

Horizontal Forecasting	MAE	RMSE	CV (RMSE) %
2 Weeks	251	339	0.61
2–4 Weeks	214	258	0.56
2–3 Months	225	294	0.63
3–4 Months	208	275	0.50
Mean-Medium term	215.6	275.6	0.56
Std. Dev	8.6	18	0.06

Table 2.6: Errors in the forecasting result in the proposed model [46].

Model	RMSE	MAE	R	SMAPE	NRMSE
Result	550.3955	458.9382	0.9624	0.0309	0.0544

Recent experimental findings in [47] suggest that the LSTM recurrent neural network yields lower forecasting errors than statistics and other machine learning-based techniques. In [48] considering an LSTM recurrent network for STLF and MTLF, to enhance the forecasting performance of STLF and MTLF, two deep learning approaches, a TD-CNN and C-LSTM networks, are presented. The TD CNN model does this by transforming the load series' temporal correlation into spatial correlation and storing it persistently. To extract the temporal correlation between the long-term sequences with less complexity, the C-LSTM method is proposed. This method creates a new short series from the original long-load series without losing any information along the way. The LSTM is then used to dynamically interact with the load series with fewer iterations, and Table 2.7 summaries the results.

Table 2.7:The performance of the proposed methods in [48].

Type	forecasting period	Algorithms	MRE %
STLF	One Weak	TD_CNN	3.3
		C_LSTM	3.3
		LSTM	2.7
MTLF	Half Year	TD_CNN	4.9
		C_LSTM	5.7
		LSTM	5.9

The RF was used in [49, 50, 51] to build models for STLF. The authors in [49] using an RF, forecast the hourly power consumption of two schools in Florida. They looked at how changing certain predictor factors affected the results. Both RT and SVR were evaluated, and their RF performance was compared to both. Using a combined performance measure of r^2, RMSE, and MAPE, they found that RF performance was superior to both RT and SVR, as shown in Table 2.8. In the study in [50] in a hotel in Madrid, Spain, they contrasted the results of using a popular ANN to predict energy usage for air conditioning with using RF. where the RMSE of 4.97 against 6.10, ANN was shown to perform somewhat better than RF. In [51] Proposed an STLF model capable of projecting time series for the following 24 hours of electricity usage, where used RF and a genuine historical dataset (from the Tunisian electricity utility) with an average inaccuracy of roughly 2.6 percent, table 2.9 compares the proposed model to other models (persistence model (PER), SVM, ANN).

Table 2.8: The performance of the proposed methods in [49].

Model	Model 1 (RT)	Model 2 (SVR)	Model 3 (RF)
RMSE	3.55	4.54	3.22
MAPE	5.21%	6.10%	4.79%
PI	0.85	1.00	0.79

Table 2.9: Comparison between the different methods in terms of MAPE [51].

	PER	RF	ANN	SVM
MAPE	5.9320	2.6899	2.9140	2.8452

Researchers in [52] tested CNN, two forms of neural networks, to see whether they might outperform regular ANN at managing time-series data. In [52] proposed a variant of CNN with a two-dimensional input for short-term load (one-day future) forecasting (consumptions from past states in one layer and meteorological and contextual inputs in another). In a study case including Algeria, they found that their approach resulted in MAPE and RMSE of 3.16 % and 270.60 (MW) of inaccuracy, respectively.

To forecasting hour ahead future electricity consumption, a technique known as NRE load forecasting was devised. Authors in [53] built an STLF, in which a method for determining a nonlinear relationship between the values of used electricity and the corresponding temperature values has been presented NRE. The nonlinear correlations between the load values were extracted using a CNN. They compared the NRE method with LSTM, CNN, and SVR. Where the MAPE is 0.38 and the RMSE is 0.95, the trials demonstrate the higher performance of the suggested NRE technique. The study in [54] created an innovative approach using local learning with SVR for energy forecasting in large data sets. Using a machine learning framework for Big Data, evaluate the state of the art in local SVR in comparison to traditional SVR and deep neural networks. On both the accuracy of its forecasting and the speed with which it could be computed, local SVR outperformed SVR and deep learning. These researchers improved forecasting accuracy while decreasing computing complexity and training time. The results of the SVR local where give MAPE - Hourly readings is 16.806 and MAPE - 15-minute readings is 19.407. In addition, based on real data from Turkey and four time-series methods, a model was developed [55] to forecasting electrical energy consumption for the short term one day: LSTM neural network, ANFIS with SC, ANFIS with FCM, and ANFIS with GP. The best MAPE values achieved during the testing phase were 4.47%, 3.21%, 2.34%, and 1.91 %for the ANFIS-GP, ANFIS-SC, ANFIS- FCM, and LSTM, respectively. As a result, the LSTM model beat all ANFIS models in general. The findings demonstrated that forecasting short-term daily time series using the LSTM technique might yield good accuracy outcomes.

2.3.2 Short-Term Load Forecasting For Small Electrical Networks

STLF for small networks is becoming increasingly critical as the penetration of distributed and renewable energy grows. Having an accurate STLF for the small grid helps resource management of both renewable and conventional resources, as well as energy economics with electricity markets. As a result of the load time chain's non-smooth and extremely unpredictable behavior in a small network. As a result, the authors in [56]–[59] built models to forecasting short-term electrical loads for small networks based on real data obtained from smart meters in buildings. [56] Used a model to forecast house electricity energy usage an hour in advance based on real data from Switzerland and deep learning techniques. They propose a DCNN based on ResNet and used GRU, ResNet, LSTM, and GCNN because the deep neural network provides an excellent ability to deal with building-level consumption patterns, regularities, and uncertainties, and the complicated non-linear linkages and correlations involved in doing so, the results showed that the proposed model (DCNN based on ResNet) has an outstanding performance in terms of accuracy and short time as shown in Table 2.10. In [57] an ensemble-based methodology for forecasting average construction consumption in France was proposed. The new framework basic learners are ensemble artificial neural networks (EANN), which are combined using multiple linear regression, where findings revealed that stand-alone ANN performed better in terms of generalization ANN-based bagging artificial neural networks (BANN). The result given RMSE (WH) = 296.3437, MAPE = 15.9396, but in BANN given the result RMSE (WH) = 309.6022, and MAPE = 16.236. [58] Suggested an LSTM-based method in which they increased forecasting accuracy by tweaking LSTM hyperparameters (Rate of learning, weight loss, momentum, and the number of hidden layers) with the Enhanced Sinusoidal Cosine Optimization Algorithm (ISCOA) and using data from India-Mumbai to forecasting long, medium, and short-term load. The results given MAE = 0.0369, MAPE=3.3159, MSE=0.0031, RMSE=0.0559 for long term forecasting, for medium-term forecasting give MAE = 0.0543, MAPE= 4.2842, MSE= 0.0077, RMSE=0.0879, and for short-term forecasting give MAE = 0.0733, MAPE= 5.1882, MSE= 0.0115, RMSE= 0.1076. A short-term electric load prediction model was built in London [59], and online adaptive RNN technology was used, a load forecasting approach that can continually learn from fresh data and adapt to changing patterns The RNN is utilized to record time dependencies, and the online aspect is accomplished by changing the RNN weights based on fresh data. The result of MAE = 0.24, 0.12 straight for 50 hours ago and an hour ago.

Table 2.10: Result Forecasting The Five Models [56].

MAE (kW)	GRU	2.57
	ResNet	1.95
	LSTM	2.6
	GCNN	1.89
	DCNN	1.43
Standard deviation (kW)	GRU	1.69
	ResNet	1.2
	LSTM	1.74
	GCNN	1.19
	DCNN	0.96

MAPE (%)	GRU	2.93
	ResNet	2.63
	LSTM	3.60
	GCNN	2.72
	DCNN	2.22
Standard deviation (%)	GRU	2.02
	ResNet	1.68
	LSTM	2.45
	GCNN	2.36
	DCNN	1.53
RMSE (kW)	GRU	2.84
	ResNet	2.57
	LSTM	3.62
	GCNN	2.57
	DCNN	2.09

According to an article in [60, 61, 62] many types of neural networks, including RNNs, CNNs, LSTM, and deep belief networks, have been used to improve the accuracy of power load estimates. In [60] method for electrical load prediction based on historical data from China has been proposed. Whereas, the proposed model is novel deep learning based on short-term forecasting, and it was compared with the SVM model. A deep CNN model was used to categorize the daily load curves. For STLF, the ANN with three hidden layers was utilized in [60]. Those take into account different environmental elements variables like heat, humidity, wind gust, and many others. Where the proposed model outperformed the SVM algorithm, where the accuracy of DLSF is 90%, and SVM= 70%. [61] Proposed a model to know the uncertainty in the electrical load profiles, based on deep learning-machine learning algorithms to predict household loads in Ireland. A novel pooling-based deep recurrent neural network (PDRNN), ARIMA, SVR, and RNN are all employed and contrasted in the study, which takes a large number of client load profiles and turns them into inputs. The result shows that the PDRNN method outperforms ARIMA, SVR, and RNN, where RMSE (kWh) = 0.4505, 0.5593, 0.518, and 0.528 respectively. [62] Proposed a model to forecast short-term electrical loads at the residential stages based on smart meter readings from the building. In [63] proposed model is CNN algorithms and compared with SVM, ANN, and LSTM algorithms, Compared to SVM, ANN, and LSTM, the outcomes favored the suggested model CNN, where RMSE = 0.677, 0.814, 0.691, 0.7 respectively.

This study [64] built a model to forecast STLF for single-family residential homes, they contrasted the LSTM model performance with the ELM, BPNN, and KNN regression to display that using the LSTM structure can reduce prediction errors by a lot, and obtained Avg. MAPE Aggregating forecasts of 8.18%, and an Avg. MAPE for individual forecasts of 44.39%.

The researchers in [65], [66] used ANN algorithms in building models for short-term electrical load forecasting since ANN algorithms deal with non-linear data. [65] For STLF in buildings, the authors propose an ANN approach to provide a robust computation using large and dynamic data to overcome the difficulty of non-linearity in combining existing data sets. The authors created and confirmed their results on a testbed home, which was supposed to be a real test facility. They built their model on the OWO newton algorithm and the Levenberg-Marquardt algorithm and achieved a r^2 is 0.91, which means the model is a perfect fitting with a rate of 90% of the anticipated variation in the predictor variables of power consumption. [66] Proposed a model for load forecasting algorithms employed a NARX neural network and SVR to predict power usage 15 minutes into the future for a day, a week, and a month, and compared the two. They evaluated the models with varied time horizons after training them with genuine data from three real commercial buildings. Based on their research, they found that the SVR was superior to the NARX neural network model. For Short-term day, weekly, and monthly forecasts, the average predicting accuracy is approximately 93%, 88-90%, and 85-87%, respectively.A model in [67] proposed to forecast short-term load forecasting for many families by employing Bayesian networks, the multivariate algorithm forecasts the next immediate household load value based on past patterns of consumption, environmental considerations, socioeconomic status, and electrical power use. Using actual data from the Irish intelligent meter project, its performance was compared to that of other forecasting systems. The findings show that the MAE (kWh) for the proposed method is 1.0085 and the MAAPE (kWh) is 0.5035, despite the fact that it provides a single prediction model for hundreds of households with wildly varied consumption habits.

All in all, predicting electrical loads in the short term helps in predicting loads for a few minutes, hours, a day, and sometimes a week will help in controlling the distribution of loads in real-time and evaluating the safety of the electrical network. However, based on the mentioned works, some researchers encountered difficulty in obtaining accurate data on the consumption behavior of consumers.

2.4 Long-Term Load Forecasting(LTLF)

Algorithms of ANN are used in [68][69] owing to their convenience in dealing with non-linear historical data, to make predictions about LTLF and MTLF. An innovative method for long-term load forecasting (LTLF) based on a rolling stochastic ARIMAX model was developed in [68] to mitigate the impact of the covid-19 pandemic on the accuracy of load projections. The suggested technique attempted to improve prediction performance by recognizing the non-smooth demand nature and recommending several future demand scenarios based on a probabilistic model. They applied machine learning models like Stochastic ARIMAX-Fixed, Stochastic ARIMAX-Rolling, ARIMAX, and ANN, the proposed model (Stochastic ARIMAX-Rolling) achieved superior performance compared to the ARIMAX model and ANN, and significantly decreased forecasting error, Table 2.11 shows the result. In [69] 2018, offered each district, an accurate long historical data load forecast based on environmental and historical data utilizing ANN and multivariate linear regression, as well as an adaptive boosting model. The results in [69] demonstrate that such models might improve forecasting accuracy in the smart grid context by taking into account suitable forecasting durations (i.e., long historical data load), as shown in Table 2.12. Authors in [70] used the NN technique and real data from the Jordanian electricity system from 2015 to 2019 and used BPNN and RBFNN the two types of FFNN. The suggested NN technique in [70] is successful in forecasting Jordan's system's yearly peak loads, with MSE = 76.2 and MAE = 7.

Table 2.11: The mean absolute percentage error (MAPE) [68].

	Stochastic ARIMAX (Fixed)	Stochastic ARIMAX (Rolling)	ARIMAX	ANN
MAPE	5.1%	4.3%	5.2%	5.7%

Table 2.12: Result proposed model in [69].

	ANN-NAEMI	MLRM	AdaBoost
MAPE (%)	4.226	4.723	0.0197
MAE	368.833	441.262	1.929

Also, models used in [71, 72, 73] are based on LSTM algorithms since they deal with long historical data, and there is a short-term storage memory for historical data. The model proposed in [71] was used to forecast MTLF and LTLF in Canada using machine learning algorithms like SVM, RF, and deep learning methods like NARX, and RNN. The results show non-linear external neural network gives the best performance and accuracy, where the R^2 is 0.96 and the MAPE is about 4-10%. A RNN model in [72] proposed to make MTLF and LTLF to forecast electricity consumption in commercial and residential buildings with an accuracy of one hour. They created and optimize DRNN models based on LSTM to generate predictions across time horizons ranging from a few months to a year at a one-hour resolution. They developed a data imputation strategy for missing values in power consumption data using the deep RNN model. The result of the proposed model are RMSE = 46.8 of testing data, RMSE = 45.8 of training data, and Pearson Coefficient (P) = 0.765. A novel approach for predicting long historical data pregnancy with hourly accuracy has been presented [73]. This model was based mostly on an LSTM-RNN cell. LSTM-RNN takes into consideration long historical data associations in time-series data for energy load demand, resulting in predictions that are more accurate. The suggested model in [73] is based on real-time data from England's new power market ISO. To train and verify the model, data were gathered over twelve years, from 2004 to 2015. Electricity demand predictions were prepared on a rolling basis for five years, from 2011 to 2015. The suggested model was shown to be quite accurate, with a MAPE of 6.54 within a 2.25% confidence range.

All in all, forecasting electrical loads in the long and medium term helps in forecasting loads from months to years. It helps in balancing the demand for electricity and scheduling maintenance and contributes to planning the growth of the network in the future. However, some studies that forecasting loads for more than two years have less accuracy because it depends on weather factors and it is difficult to forecasting them for some years. Some researchers [74] encountered problems in obtaining atmospheric data to be entered as factors in forecasting electrical loads. In addition, some models that were built to forecasting long historical data loads take a large amount of time during training, and this depends on the algorithms used by researchers.

2.5 Summary of Results

Table 2.13 shows a summary of the studies that have been conducted in the literature review and the results they have reached in the forecasting of electrical loads over the short, medium, and long terms.

Table 2.13: A Summary of the literature review.

Reference	Type (STLF, MTLF, LTLF)	Algorithms	Result	Application level (Country, Building)	Location
34	STLF	LSTM, CNN, XGBoost, RBFN	CNN with LSTM was less MAE, less RMSE, and less MAPE compared with other algorithms.	Country	Bangladesh.

			MAE=304.8, RMSE= 429.8, MAPE%= 4.23.		
35	STLF	NN- double layers, NN-PSO-single layer, NN-GA- single layer, NN-EHO- single layer	A two-layer NN is the optimum for load prediction. NN- double layers PE %=0.3769,RMSE=101.355, NN-PSO-single layer PE % =2.4301 ,RMSE=261.0891, NN-GA- single layer PE % = 10.418, RMSE= 417.558, and NN-EHO- single layer PE % = 19.840, RMSE= 539.4425.	Country	Jordan.
36	STLF	NN with PSO	MAPE = 0.0338, MAE = 0.02191.	Country	Iran.
37	STLF	CNN, LSTM-GRU gate	MAE=66.12, MAPE%= 0.37.	Country	North America.
38	STLF	CNN, packing regression, and MTCN	The MTCN has the best performance R^2= 0.96, and MAPE=1.8.	Country	China
39	STLF	CNN, LSTM, MLP, ANN	The CNN has the best performance, where MAE = 3.93, and RMSE= 4.67.	Country	America
40	STLF	NN with FFGWO, LR,	NN with FFGWO had the best prediction accuracy and less error was obtained, where MAE = 221.28, and RMSE = 277.91. LR gave MAE = 271.07, and RMSE = 342.27.	Country	Jordan
41	STLF	CNN	MAPE 19.26% and RMSE 0.1666.	Country	Public
42	STLF	EMD-GRU-FS	Accuracy on four data sets was 96.9 %, 95.31 %, 95.72 %, and 97.17 %, consecutively.	Country	Public
43	STLF	LSTM with EMD	MAPE= 2.6249 % in the winter and 2.3047 % in the summer.	Country	Public
44	STLF, MTLF	NN, LSTM with GA	The LSTM with GA gave the best performance, where MAE= 251, RMSE= 339, and RMSE% = 0.61.	Country	France
45	STLF	VMD, LSTM with optimizer BOA, SVR, LR, RF, and EMD-LSTM	The LSTM with optimizer BOA gave the best, where MAPE is 0.4186% and R2 is 0.9945.	Country	China
46	STLF	Decoder architecture-based encoder-decoder network with Bayesian optimization and GRU gate.	R= 0.9624, SMAPE= 0.0309, NRMSE= 0.0544.	Country	American
48	STLF, MTLF	TD_CNN, C_LSTM, LSTM	The TD_CNN is the best where MRE%= 2.7 for STLF, and MRE%= 4.9 for MTLF.	Country	China – Hangzhou.
49	STLF	RF, RT, SVR	The results demonstrate that RF performance outperformed RT and SVR, where MAPE = 4.79%, and RMSE= 3.22.	Country	North-Central Florida
50	STLF	ANN, RF	The ANN performed marginally better than RF with RMSE of 4.97 and 6.10, respectively.	Building	Madrid
51	STLF	RF, PER, ANN, SVM	The results demonstrate that RF performance outperformed PER, ANN, and SVM where MAPE 2.6899, 5.9320, 2.9, 2.8 respectively.	Country	Tunisian
52	STLF	NARX and ANN	MAPE and RMSE of 3.16 % and 270.60, respectively.	Country	Algerian
53	STLF	NRE, LSTM, CNN, and SVR	The superior performance of the proposed NRE method where MAPE is 0.38 and normalized root MSE (NRMSE) = 0.95.	Country	Public

54	STLF	SVR based on NN	MAPE - Hourly readings is 16.806 and MAPE - 15-min readings are 19.407.	Country	Public
55	STLF	LSTM, ANFIS with (SC, ANFIS with FCM, and ANFIS with GP.	The ANFIS-GP obtained a MAPE of 4.47%, the ANFIS-SC of 3.21%, the ANFIS-FCM of 2.34%, and the LSTM of 1.91% throughout the testing phase.	Country	Turkey
56	STLF	DCNN based on ResNet and GRU gate, LSTM, CNN	The DCNN based on ResNet is the best, since MAE= 1.43, MAPE%= 2.22, RMSE= 2.09.	Building	Switzerland
57	STLF	EANN, BANN	EANN is the best, where RMSE = 296.3437, MAPE = 15.9396. In BANN given the result, RMSE = 309.6022, and MAPE = 16.236.	Building	France
58	STLF, MTLF, LTLF	LSTM, ISCOA	MAE = 0.0369, MAPE=3.3159, MSE=0.0031, RMSE=0.0559 for LTLF. For MTLF give MAE = 0.0543, MAPE= 4.2842, MSE= 0.0077, RMSE=0.0879. For STLF give MAE = 0.0733, MAPE= 5.1882, MSE= 0.0115, RMSE= 0.1076.	Building	India-Mumbai
59	STLF	RNN	MAE = 0.24, 0.12 straight for 50 hours ago and an hour ago.	Building	London
60	STLF	DLSF, SVM	The DLSF model outperformed the SVM algorithm, where the accuracy of DLSF is 90%, and SVM= 70%.	Country	China
61	STLF	PDRNN, ARIMA, SVR, and RNN.	The PDRNN method outperforms ARIMA, SVR, and RNN, where RMSE (kWh) = 0.4505, 0.5593, 0.518, and 0.528 respectively.	Building	Ireland
63	STLF	CNN, SVM, ANN, and LSTM.	The superiority of the proposed model CNN over SVM, ANN, and LSTM where RMSE = 0.677, 0.814, 0.691, 0.7 respectively.	Building	Public
64	STLF	LSTM with ELM, BPNN, KNN,	The LSTM with ELM is the best where MAPE = 8.18%.	Country	China
65	STLF	ANN based on the Levenberg Marquardt and OWO newton algorithms	The model is a perfect fitting with a rate of 90% of the variance in the power consumption variable predicted from the independent variable.	Building	Public
66	STLF	NARX, SVR	The SVR outperformed the NARX neural network model, for the day ahead, a week ahead, and a month ahead forecasting, the average predicting accuracy is approximately 93%, 88-90%, and 85-87%, respectively.	Country	Public
67	STLF	NN with Bayesian networks	MAE is 1.0085, and MAAPE is 0.5035.	Building	Irish
68	LTLF	Stochastic ARIMAX-Fixed, Stochastic ARIMAX-Rolling, ARIMAX, and ANN.	The Stochastic ARIMAX-Rolling achieved better results than the Stochastic ARIMAX (Fixed), ARIMAX, and ANN. Where MAPE=4.3%, 5.1%, 5.2%, 5.7% respectively.	Country	Public
69	MTLF, LTLF	ANN-NAEMI, MLRM, AdaBoost	The AdaBoost achieved the best performance compared with ANN-NAEMI, and MLRM where MAPE %=0.0197, 4.226, 4.723, respectively.	Country	Public

70	LTLF	BPNN and RBFNN	MSE = 76.2 and MAE = 7.	Country	Jordan
71	MTLF, LTLF	SVM, RF, NARX, LSTM	The results show NARX gives the best performance and accuracy, where the R^2 is 0.96 and the MAPE is 4-10%.	Country	Canada
72	MTLF, LTLF	RNN based on LSTM.	RMSE = 46.8 for testing data, RMSE = 45.8 for training data, and Pearson Coefficient (P) = 0.765.	Building	Public
73	LTLF	RNN based on LSTM	MAPE is 6.54.	Country	England

2.6 Conclusions

In this chapter, a compressive literature review was conducted, it began with a discussion of several publications connected to STLF forecasting, whether at the network level in regions or at the level of residential buildings.The different algorithms were discussed and presented in forecasting such as LSTM, SVM, RF, CNN, ANN, and SVR. Moreover, the forecasting of long historical data electrical loads was reviewed in terms of algorithms used in forecasting and the comparison between the algorithms.

As noted, there is no forecast of electrical loads in the State of Palestine, since forecasting for each country is different from the other because the terrain and climatic conditions differ from one country to another as well as the population density and number. In addition, no interfaces are connected to the forecasting system to monitor the loads in the literature reviewed.

Our work will focus on forecasting short-term electrical loads based on real data in Palestine and add new data online from SCADA to the training dataset directly. Also will create a fine-tuned machine-learning model to forecasting short-term electrical loads with the highest accuracy and least error rate, which will help to solve the problem of power outages in Palestine and save time and cost.The research conducted in this thesis is aiming to answer the following questions: 1- What is the best model that gives the highest accuracy? 2- What is the best optimizer that can be used that gives the best performance and accuracy? 3- What are the days of highest consumption? 4- How is the distribution of electricity consumption during the hours of the day? 5- What are the peak-hours in terms of consumption?

Chapter Three

Exploratory Data Analysis

3.1 Introduction

In this chapter, all exploratory data for electrical load data were extracted and analyzed. Exploratory data analysis is an examination of the many features, correlations, and hidden patterns found in electrical load data. Several methods have been used to analyze and explore the data such as autocorrelation, box plot, and line plot. The ability to visualize and investigate the interrelationship of different variables and to unearth previously unseen patterns is a key feature of EDA that is crucial to the creation of time series forecasting models. In order to validate and formally model the forecasting, an EDA must first be conducted [75]. The following topics will be covered in this chapter: 1- data collection, 2- correlation, 3- electrical demand behavior analysis, 4- time series analysis for electricity loads, and 5- conclusion.

3.2 Data Collection

The data used in this thesis were obtained from Tubas Electricity Company - Palestine. All loads are stored through the SCADA program in a database, and every minute the system enters the new reading into the databases. Knowing that in this thesis, one connection point was taken out of two connection points (points between the electricity company and the Israeli side). The dataset serves an area that contains agricultural wells - the Tubas area, as the loads depend heavily on agricultural wells, especially with high temperatures; the electrical loads are high, because of the agricultural wells that work for between 8-15 hours per day. This data contains 348,303 records from the date September 1, 2021, to the date of Jun 07, 2022. The dataset of the period before September 2021 was not taken because there is no temperature feature in the data because the Electricity Distribution Company did not install a sensor to measure temperatures during that period. The most important features found in this data are (date and hour, temperature, consumption, energy, and others). The dataset contains 360,748 rows and 8 columns as shown in Table 3.1, which represents the first five records of the data. This dataset will also be discussed in the next chapter [Chapter 4- section 4.2] and the data processing will also be discussed in the next chapter [Chapter 4- section 4.3].

Table 3.1: First five records for dataset samples.

Date	Temperature ºC	Hour	Weekday	Week	Month	Year	Current
2021-09-01 00:00:54	31.0	0	3	35	9	2021	284.10560
2021-09-01 00:01:55	31.0	0	3	35	9	2021	279.18033
2021-09-01 00:02:55	31.0	0	3	35	9	2021	278.64350
2021-09-01 00:03:56	31.0	0	3	35	9	2021	280.11516
2021-09-01 00:04:56	31.0	0	3	35	9	2021	280.37660

3.3 Correlation

The linear link between two or more variables may be measured using a statistical technique called correlation. One variable may be forecast from another via the use of correlation. The theory behind utilizing correlation to select features is that useful variables will have a strong correlation with the result. In addition, there has to be a correlation between the variable and the endpoint but no correlation between the variables themselves. The heat map assists in understanding the correlation ratio between the features to know whether there is enough connection to construct a regression model to forecasting the short-term load to examine the link between the dataset components. The heat map was created with the python matplotlib and seaborn module, which calculates the Pearson correlation coefficient (r) between the components using the following equation.

$$r = \frac{\sum(x_{i-\bar{x}})(y_i - \bar{y}\,)}{\sqrt{\sum(x_{i-}\bar{x})^2}\,\sum(y_i - \bar{y})^2} \qquad [76] \qquad 3.1$$

Where

r = Pearson correlation coefficient

x_i = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

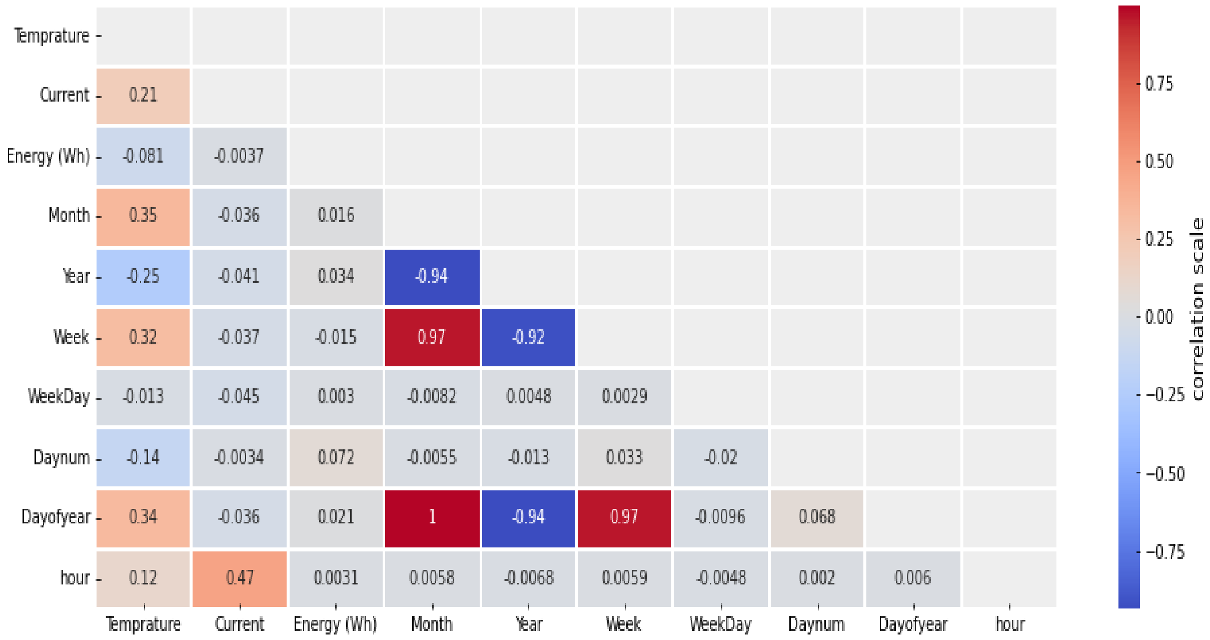

Figure 3.1: Heat map for correlation dataset features.

Figure 3.1 shows the correlation between the elements with each other, and we note that there are positive relationships between some elements such as (week and month, and also the week and number of days of the year) and negative relationships like (year and month, year and day of the year). As shown in the figure, the relationship between temperature and consumption of current (energy) is approximately 0.21, because during the winter there is no consumption of electricity by agricultural wells, this is because the area served by the company is an agricultural area in which there are many agricultural wells. In the summer, consumption is high because there is no rain to irrigate the crops. In addition, the relationship between electricity consumption and the hour is equal to 0.47 because the loads are highest in the morning during the working hours of the institutions, and at night, they are the lowest.

To find the relationships between the days of each month for the consumption of electricity figure 3.2 show it.

Figure 3.2: Heat map for demand to show correlations between days and months.

Figure 3.2 shows the relationship between days and months for electricity consumption. As noticed that in September, the consumption had the highest values compared to the remaining months, as Wednesday had the highest consumption where the result is 1.8 x 106, and Friday was the lowest consumption day is 1.2*106. The followed by January month was his second rank of consumption. Months November and December had the lowest consumption. Where it can be seen that in general, consumption is high from Sunday to Thursday, and it is few on Fridays and Saturdays. The correlation in consumption between January and May was almost the highest with the days of the week, and this indicates a decrease in consumption in this period because it is a period of winter and rain and the agricultural wells are operating temporarily.

3.4 Electrical Demand Behavior Analysis

To find out the days and hours during the day and the days of the months that have the highest electricity demand. Figure 3.3 - 3.6 shows this, to find the minimum, first quartile (Q1), median, third quartile (Q3), and maximum for each type. It also helps in forecasting the times, months, and hours when the loads are overloaded to take the necessary measures to avoid the occurrence of problems. It also helps in forecasting the times when the loads are the least, in order to reduce the losses in the network.

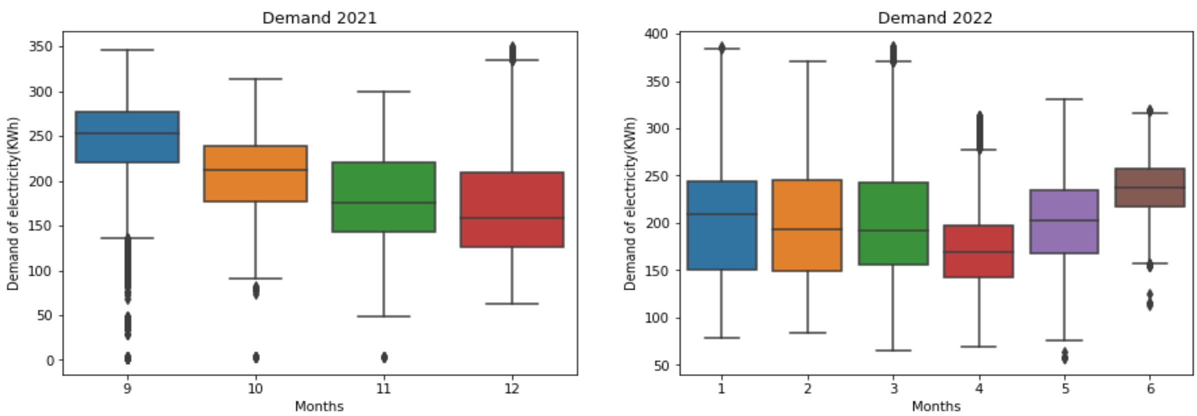

Figure 3.3: Box plot demand based on months from 2021 to 2022.

Figure 3.3 shows the electricity consumption from September 2021 to June 2022, it can be seen that the consumption in September was the highest among the months since during this month temperatures are high. All citizens operate air conditioners, and all agricultural wells are working to irrigate crops. As for December, It is the least in consumption, because the cans of agricultural wells are not working due to the irrigation of crops from rainwater. This helps how to deal with data and forecasting loads, especially in the summer. Moreover, most importantly, forecasting in the summer season helps to take the necessary technical measures without causing problems in the network due to the large consumption of the network. It can also be helping to obtain electricity from alternative sources. This is helping to focus on forecasting electrical loads in the summer to avoid the season or expected problems due to overloads.

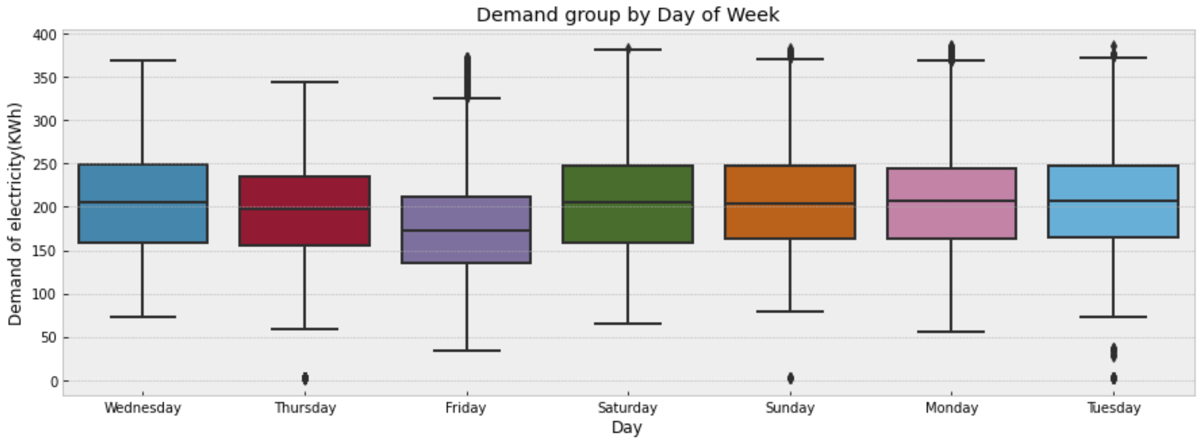

Figure 3.4: Box plot demand based on the day of the week.

Figure 3.4 shows the electricity consumption from the September 2021 month to June 2022 group by day of the week. It can be seen that the consumption on Wednesday is the highest between the days of the week where q1 is 158.024 and q3 is 249.422. Friday is the lowest day of the week because it is the weekend holiday, and companies and factories are closed on this day where q1 on Friday is 135 and q3 is 211. This analysis helps to focus on forecasting electrical loads during working days for employees and schools, and on Friday there are points or abnormal loads because consumption is low on this day. This is also reinforced by focusing on forecasting loads on days with excessive loads to avoid electrical faults and working on the correct distribution of electricity based on days.

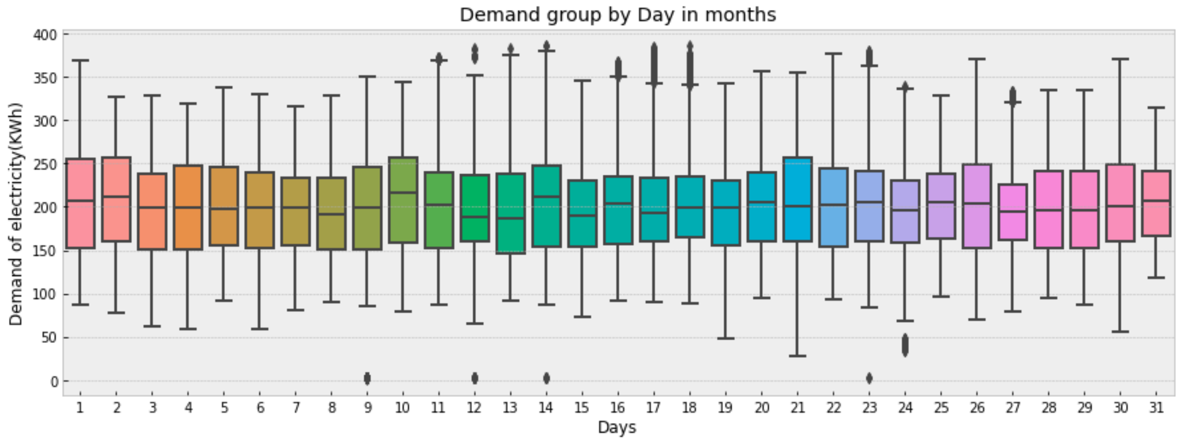

Figure 3.5: Box plot demand based on the day of the month.

Figure 3.5 shows the distribution of electricity consumption from the September 2021 to June 2022 group by day of the month, It can be seen that the consumption on the 10th and 14th are the highest between the days of the month where the max value is 386.2813 and the Median= 211. Moreover, the 7th is the lowest day of the month where the max value is 315.2 KWH and the Median is 199.

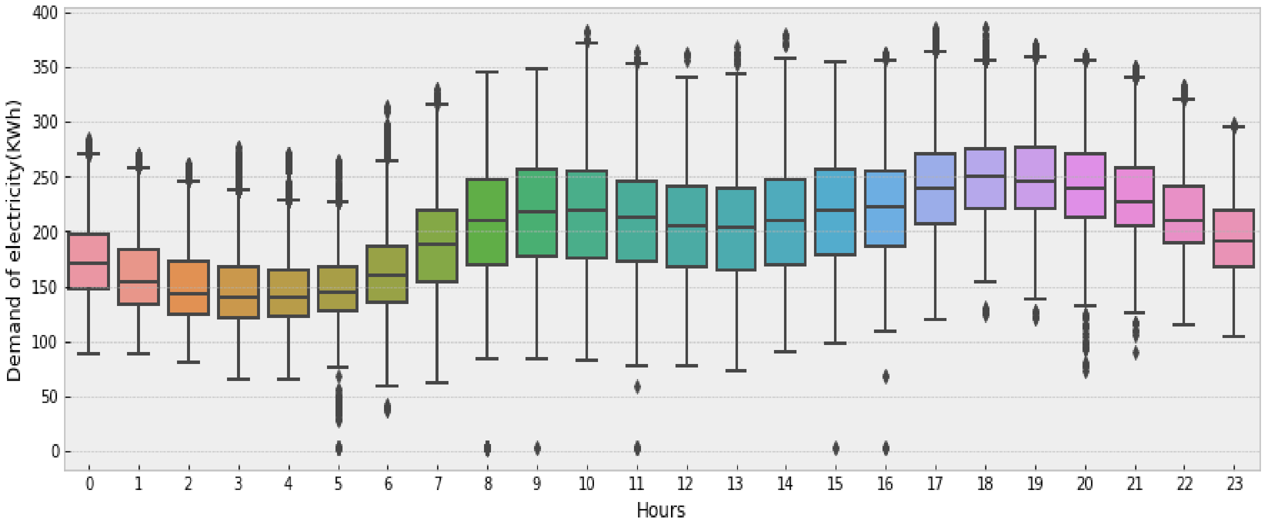

Figure 3.6: Box plot demand based on the hour of days.

Figure 3.6 shows the consumption of electricity based on the hours during each day. It can be seen that in the period between 6 am, the consumption of electricity begins to gradually increase to 8 pm because it is almost the beginning of the work of factories, companies, and agricultural wells. Then as noticed, the consumption begins to gradually decrease in the available night hours and the early morning hours, because it is the sleeping period for families, and the shops are closed. It has also been mentioned that the loads start to increase from six in the morning until eight in the evening, where the highest values of the loads in this period are between (314-360) kwh. During the night and early morning hours, the loads are between (260-299) kwh.

3.5 Time Series Analysis For Electricity Loads

To identify the demand for electricity on a daily, weekly, monthly, and during the day. Figures 3.7 – 3.10 show this, which this is helping to identify the nature of the data, whether is it seasonal, recurring, or random. moreover, it shows some statistical measures to show the distribution of some figures of the dataset.

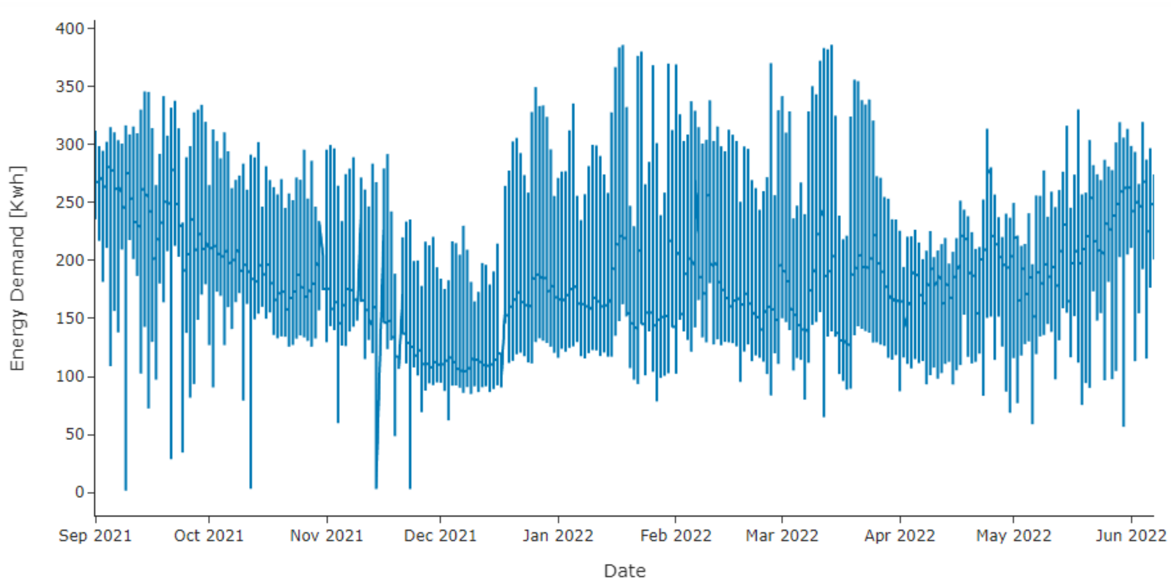

Figure 3.7: Power Demand (KWM) over time [Sep-2021, Jun-2022].

Figure 3.7 shows the electricity consumption from the beginning of September 2021 to June 2022 every minute. It can be seen that the consumption is variable and noticed that the highest value is 400 and the lowest value is zero, and this indicates that there is a disconnection in the electric current at that time. Also, conclude that the consumption from December 2021 to the beginning of February 2022 was low, because it was the beginning of the winter period; in addition to that the agricultural wells were operated less than the summer period, because the area served by the company is an agricultural area in which there are many agricultural wells. In the summer, consumption is high because there is no rain to irrigate the crops.

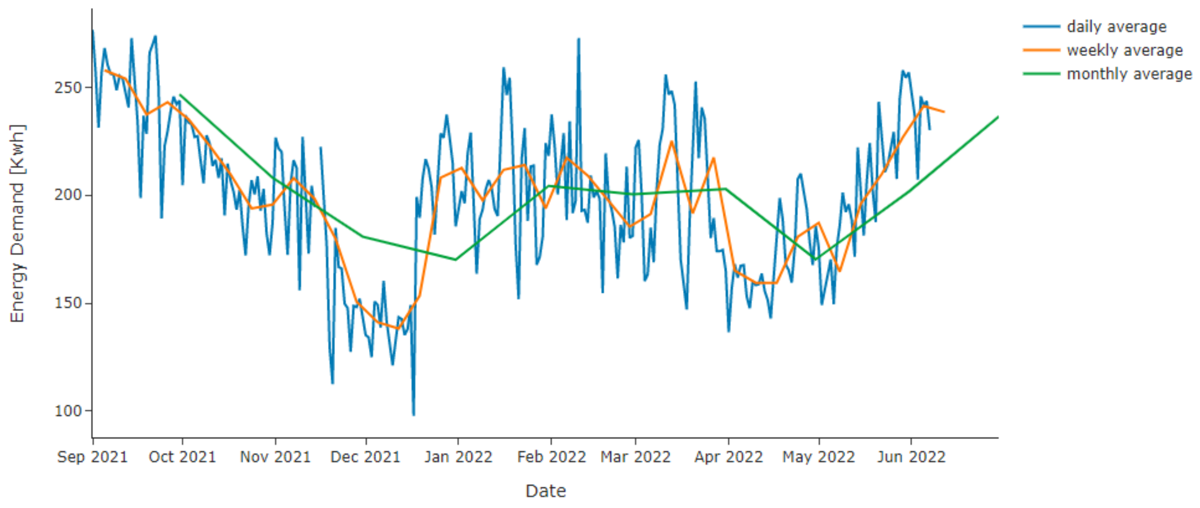

Figure 3.8: Tubas electricity demand.

Figure 3.8 shows the electricity consumption from the beginning of September 2021 to June 2022 in the form of (daily, weekly, and monthly averages). It can be seen from the monthly average that consumption from September 2021 starts decreasing until February 2022, because this period is in the winter season and the summer season is over. In addition, especially between December 2021 to January 2022, it was the least possible because it was the rainy period and agricultural wells were almost stopped because the crops were watering from rainwater. During the period between February 2022 and May 2022, consumption was virtually constant, after which consumption began to rise with the beginning of the summer season. With regard to the distribution of loads on a monthly basis, it is clear that the loads have changed from September 2021 to the beginning of January 2022, then they start to rise, and this indicates that the temperature is the largest determining factor in relation to electrical consumption. In addition, the distribution of the data on a weekly and almost daily basis is close. It is clear that the loads are equal in the period February 2022 to April 2022, and with the beginning of June 2022, the loads begin to rise, because the temperature begins to rise and the winter is interrupted, there is no rain, which leads to the operation of agricultural wells. Table 3.2 show the statistical analysis for Figure 3.8.

Table 3.2: Descriptive measures for electrical load dataset.

	Daily	**Weekly**	**Monthly**
Standard Deviation	35.59	30.15	25.13
Mean	199.013	200.51	202.18
Median	200.36	198.31	202.25

Table 3.2 shows the description standards for electrical load data on a daily, weekly, and monthly basis. Mean, median, and standard deviation was found. The mean for daily electrical loads is 199,013 kWh, the weekly average is 200.51 kWh, and the monthly average is approximately 202.18 kWh. The mean is close to the same loads. As for the standard deviation, it is clear that the distribution of loads daily is 35.59 kWh away from the average, which is the most dispersed from the arithmetic mean, and the lowest is the standard deviation of monthly loads of 25.13 kWh. As for the median, the average daily electrical load is 200.36 kWh, the weekly is 198.31 kWh, and the monthly is 202.25 kWh. Therefore, it is clear from the previous explanation that the dispersion of the daily electrical loads from the arithmetic mean is the highest.

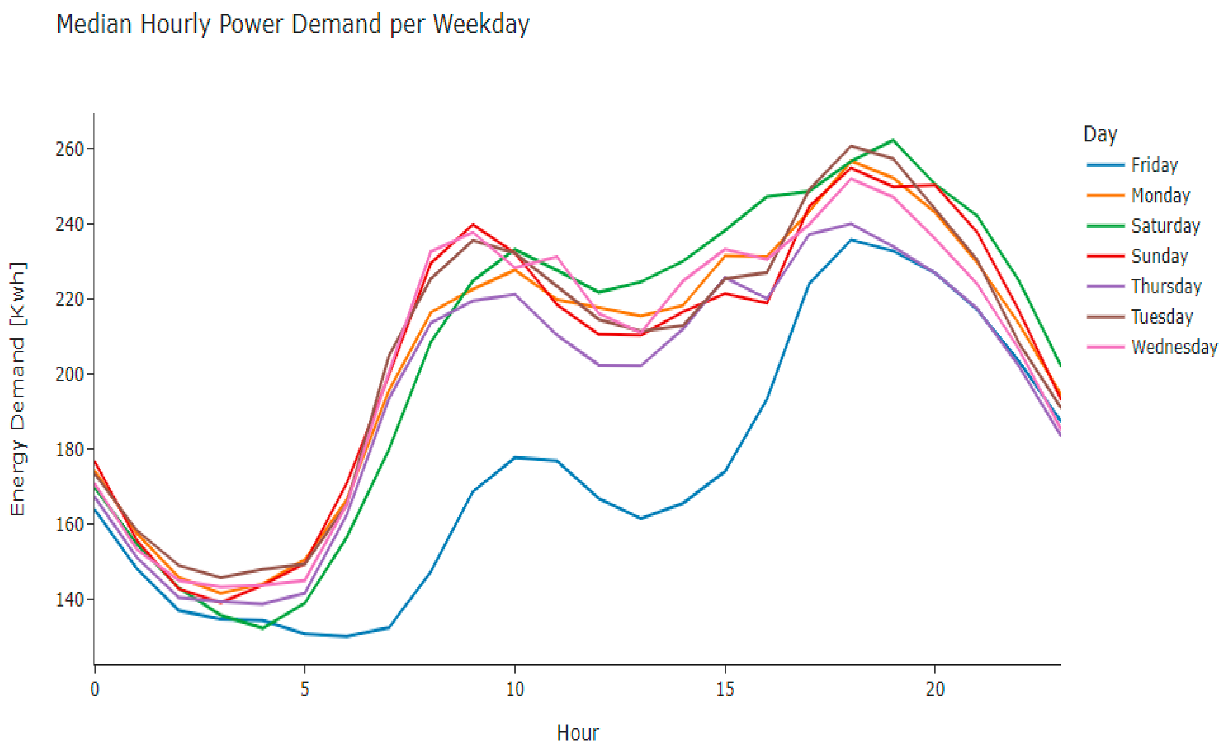

Figure 3.9: Hourly Power Demand per Weekday.

Figure 3.9 shows the relationship between the days of the week and the hours during the day. It can be seen that Friday having the lowest demand for electricity because it's a weekend day, and on the rest of the days, the demand is almost uneven between them. In general, the demand for electricity starts decreasing from 20:00 to 6:00 am. where the demand is between (240 - 350) and then the demand begins to increase to reach the peak between hours 8-19 where the demand is between (350-390) KWh every day without the weekend day where the max load is 233 KWh and the min load is 86 KWh.

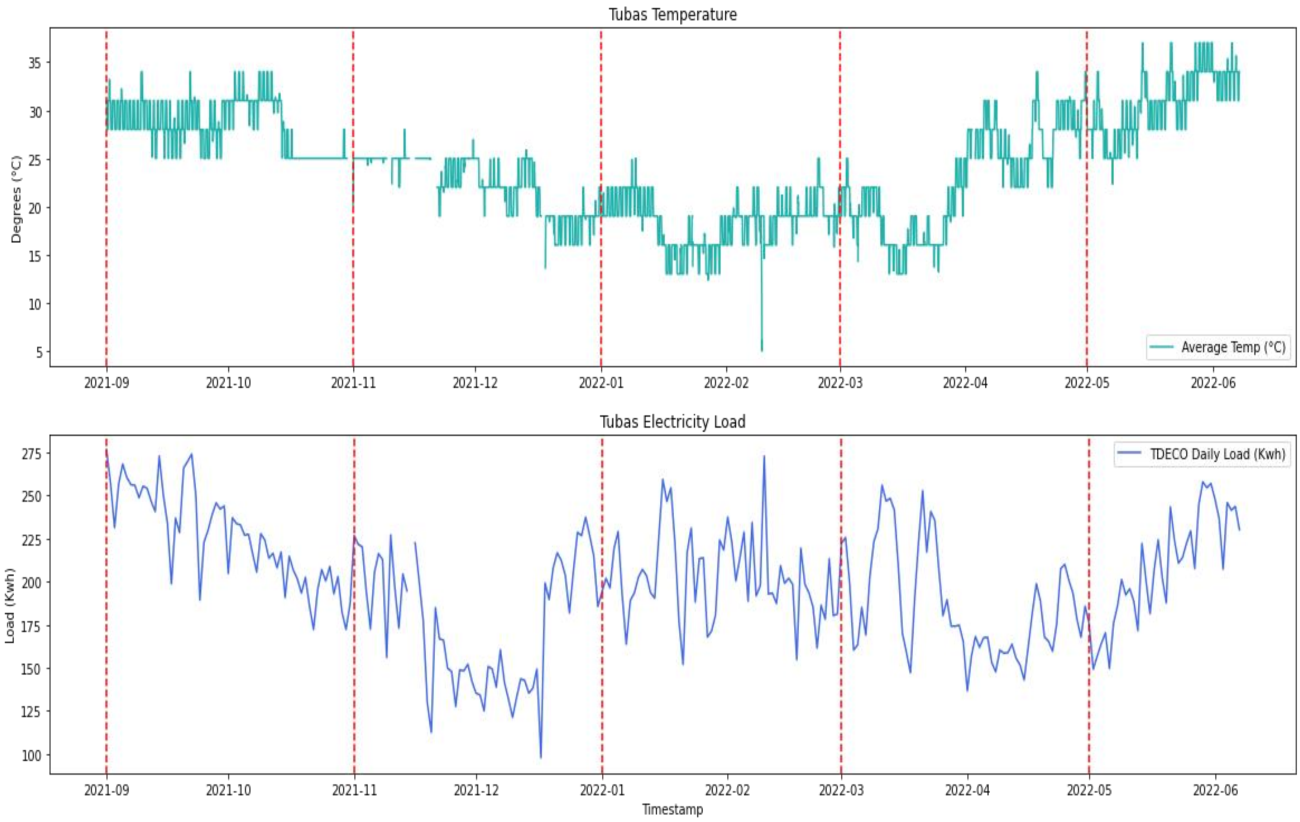

Figure 3.10: The relation between temperature in degrees and the demand power in Kilowatts hours.

Figure 3.10 shows daily temperatures and daily electricity consumption. It can be seen from September to November, temperatures ranged between (25 °C and 33 °C), and the demand for electricity during this period was high, especially from the beginning of September to mid-October because the temperature was high. From mid-October to the beginning of November, the consumption was low because the temperature was low during this period. After all, the air conditioners were almost turned off. In the period between November and the beginning of January, temperatures ranged between 25-15 °C, and the demand for electricity during this period was minimal because all air conditioners were turned off and most agricultural wells stopped because farmers depend on rainwater to irrigate crops. Moreover, during the period between January and February, the temperatures are between (20 °C and 5 °C), as the consumption was high because the air conditioning was working and the rains were almost low, forcing farmers to operate agricultural wells. Then, from March to June, temperatures begin to rise gradually, and accordingly, the demand for electricity increases or decreases according to the temperature.

3.6 Conclusion

In this chapter, an exploratory data analysis was performed, which allows us to focus on data patterns and decide how to utilize machine learning to extract knowledge from the data. After visualizing the data, It can be seen that when the temperatures are high, the demand for electricity increases, and when the temperatures are medium (15 ^{o}C and 25 ^{o}C), the electricity demand is affected, and at low temperatures (less than 15 ^{o}C), the demand for electricity increases. In addition, the peak electricity consumption is from 6:00 to 20:00. Further, the lowest consumption is on weekends (on Fridays). Following this exploration, in the next chapter, the methodology used to forecast the electric load and dataset will be presented and discussed in detail.

Chapter 4

The Proposed Methodology

4.1 Introduction

This chapter explains the proposed method, which aims to forecast short-term loads based on deep learning and machine learning algorithms in a real dataset containing loads and independent variables. It begins with composing the data set and describing preprocessing steps. The chapter then explains the standard models: Deep learning models to forecast short-term load based on historical data. The chapter then explains the hyperparameters tuning for machine learning models such as (optimizer, activation function, learning rate, number of epochs, batch size, and number of hidden layers). Finally, it shows the metrics used to measure the performance of models. Regression was implemented using Python 3 with the Pandas, Seaborn, Sklearn, Matplotlib, TensorFlow, and NumPy libraries; Used a 2.6GHz intel core i5 processor platform with 8GB of RAM running on a Windows 10 desktop to implement this approach.

4.2 Datasets

The electrical data is collected and assessed in accordance with the real-time supervisory control and data acquisition (SCADA). The load on the same denomination day is determined by looking at both the load at the hour before and the current load. The continuous ratings of all load-consuming appliances that are connected to the system are added together to get the cumulative connected load. At the same time, the biggest demand placed on the system is determined by adding up all of the requests that have been made within a certain time frame (daily, monthly, and yearly). Tubas District Electricity Company (TDECO) in Palestine provides actual load statistics for a year. The basic input data set is comprised of historical consumption, date, day, and temperature information. STLF is carried out utilizing load data from 1 September 2021 to 7 June 2022. Every quarter hour, the load data was recorded. Many factors influence electrical use, including seasonality, climate, electricity pricing, and extraordinary occurrences.

4.3 Data Preprocessing

Data preprocessing is a key stage before developing and training the machine-learning model to use the data; without preprocessed data, the machine-learning models may not operate as effectively as necessary, resulting in a miss and poor outcomes.

Depending on the nature of the raw data, different preprocessing sub-steps may be used [43], data normalization, and removing highly correlated features were used in this thesis, as well as removing features with little correlation with the targeted feature, outliers removal, and an evaluation to check the null values in the original data. The techniques will be described in detail in the following sections.

4.3.1 Data Normalization

Data normalization is a preprocessing technique used to prevent some features from dominating all other features; data normalization aims for features with the same scale to be of equal importance; there are many types of data normalization methods, including standardization and max-min normalization [77].

The range of all features was standardized to be between [0-1] in this study, and the max-min normalization technique was used to conduct a linear transformation on the data, with the max-min normalization method determined using equation 4.1 as described below [78].

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad 4.1$$

Where x' is the normalized value, x is the original feature value, x_{min} is the minimum value of the feature and x_{max} is the maximum value of the feature.

4.4 Machine Learning Algorithms

In this part, will provide a quick overview of the chosen three ML algorithms. The algorithms were chosen based on their popularity and performance in prior studies. Although sharing the same goals, an extensive range of ML regression algorithms has quite distinct mathematical models, strengths, and drawbacks.

The regression forecasting continuous values for outcomes by leveraging the relationship between two or more quantitative variables. There are two sorts of variables: the dependent variable (response variable) and the independent variable (predictor variable). The regression method explores the relationships between the elements throughout the regression process so that the dependent variable value may be predicted by the independent variable. We used LSTM, RNN, and GRU for the regression model to forecast the power consumption, figure 4.1 shows the proposed method for regression.

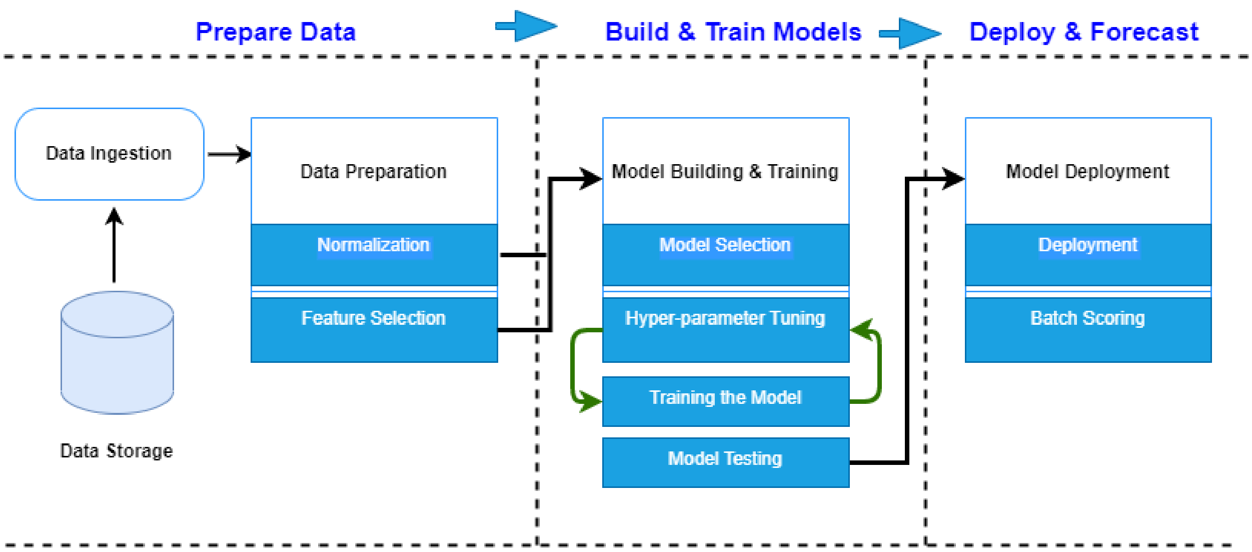

Figure 4.1: Workflow Load Consumptions Forecasting models.

4.4.1 Recurrent Neural Network Model

Time-series analysis is a specialty of recurrent neural networks (RNNs), which have found effective applications in fields as diverse as speech recognition, machine translation, and image captioning [82], [83]. Rather than revealing the contents of previously recorded time steps, RNN analyses incoming sequence/time-series data by a unique vector for each level.

Figure 4.3 shows an RNN utilizing the basic recurrent unit. The W_h, W_y, and U_h matrices, respectively, include input weights, output weights, and recurrent weights. b_h and by our biases, whereas $act_h(z)$ and $act_y(z)$ are activation functions. To maintain the observed information, a hidden state vector H_t with the same length as the input variables is constructed. In the first time step, its values are set to zero.

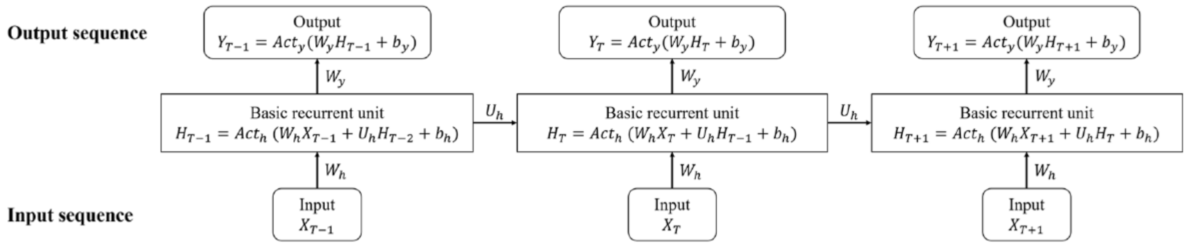

Figure 4.2: Schemata of Recurrent Neural Network.

The hidden state vector at time step T-1 (H_T -1), along with the input data at time step T (X_T), will be utilized to determine the hidden state at time step T(H_T). In theory, RNN may maintain information observed in prior time steps and has a stronger potential for learning temporal correlations or dynamics than traditional ANN.

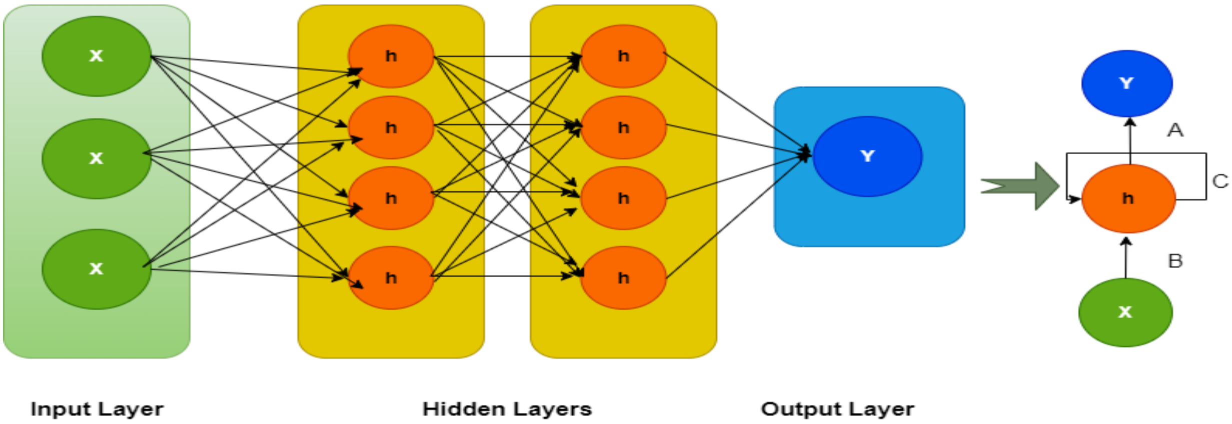

Figure 4.3: The simple recurrent neural network structure.

Figure 4.4 shows, "x" represents the input layer, "h" represents the hidden layer, and "y" represents the output layer. The networking variables A, B, and C were employed to enhance the model's outputs. The current input is a mixture of input at x(t) and x(t) at every particular time t. (t-1). At each particular time, the output was retrieved back to the network to enhance it.

However, the capacity of an RNN to capture long historical data temporal correlations is typically restricted because of the issue of gradients that either disappear or explode, which means that the model becomes untrainable as the number of recurrent operations increases [84]. Several RNN versions have been proposed over the last few decades to address this issue. The LSTM units [84] and GRU [85] are the most successful and extensively utilized methods. For the purpose of data processing, LSTM creates a novel cell state (learning what data in a sequence is important to keep). In other words, at each time step, the output is calculated using the current input data, the hidden state, and the cell state. To help fix the issue of vanish or burst gradients [86], the LSTM permits the reinjection of previous information at a later time by choosing the proportion by which the past information is allowed to impact the current information. A simpler alternative to the LSTM, the GRU only needs an update gate and a reset gate, which results in increased processing efficiency at the cost of somewhat lower model accuracy. For this study, will use LSTM and GRU basic recurrent units to develop and evaluate cooling demand forecasting models for the next 24 hours.

4.4.2 Long Short-Term Memory Model

A long short-term memory network (LSTM) is a sort of temporal cyclic neural network [81] that is specifically designed to address the large-tersssssm reliance issue that exists in a genera Recurrent Neural Networks (RNN). Memory units replace the hidden layer neurons of a standard RNN network in an LSTM network. The memory unit's architecture, which contains the input gate, forgetting gate, and output gate, can cause the networks to erase erroneous data or keep critical data at each time step.

Because of its capacity to learn temporal correlations, an LSTM recurrent network has become one of the best candidate networks in various domains like language translation and speech recognition. Such timing correlates are common in power consumption loads because they are based on inhabitants' behavior, which is difficult to understand and forecast. In the instance of electrical load forecasting, the LSTM network is intended to extract the phases of the loads from the patterns of the incoming power consumption profile, then store these states in memory, and lastly forecast based on the learned knowledge [64]. Figure 4.2 shows the construction of an LSTM cellblock. The computation procedure is depicted in Equations (4.2) - (4.7).

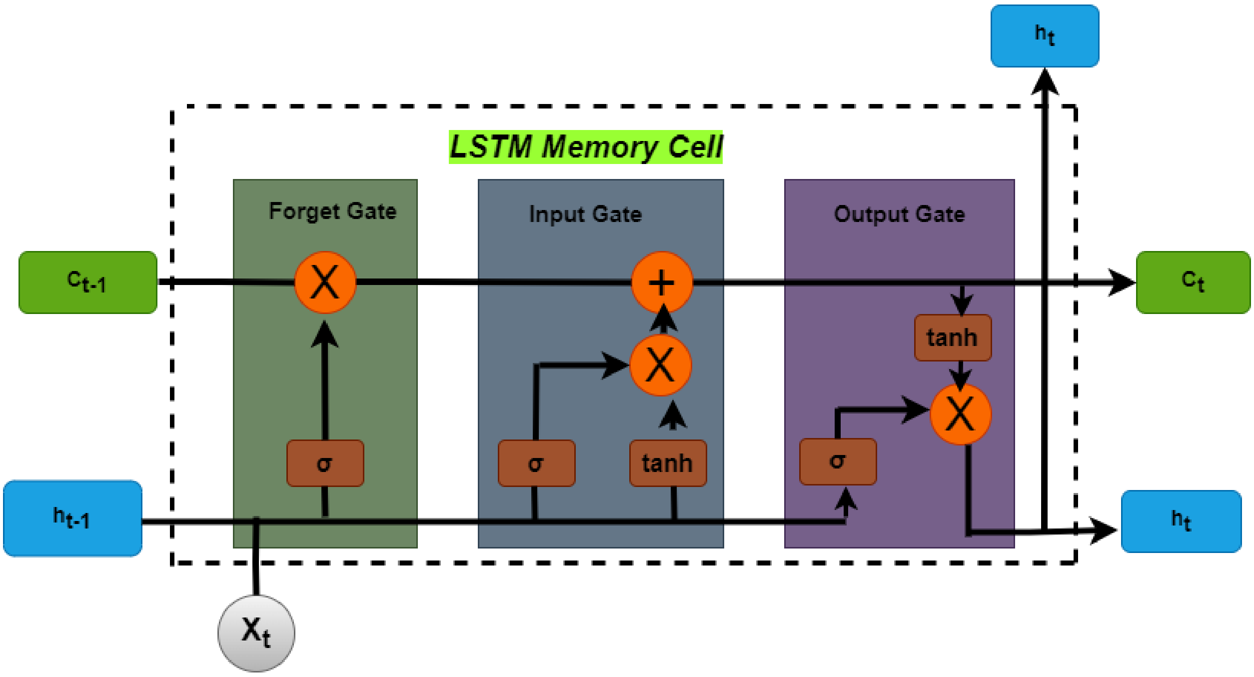

Figure 4.4: LSTM block structure.

$$f_t = \sigma(W_f|s_{t-1}, x_t| + b_f) \qquad (4.2)$$

$$i_t = \sigma(W_i|s_{t-1}, x_t| + b_f) \qquad (4.3)$$

$$\overline{C_t} = \tanh(W_c|s_{t-1}, x_t| + b_c) \qquad (4.4)$$

$$C_t = f_t C_{t-1} + i_t \overline{C_t} \qquad (4.5)$$

$$o_t = \sigma(W_o|s_{t-1}, xt| + b_o) \qquad (4.6)$$

$$s_t = o_t \tanh(c_t) \qquad (4.7)$$

First, the forget gate f_t is computed to assess how much information the current memory cell has forgotten from the prior cell state C_{t-1}, The memory cell's input is the current input x_t paired with the previously hidden state s_{t-1}, W_f is the forget gate's weight matrix. i_t Is determined by the input gate how much information from the input signal may be transferred into the global cell state. W_i is the input gate's weight matrix.

The current cell state $\overline{C_t}$ is then determined, and the global cell state C_t is updated based on the input and forget gates. Lastly, the output gate o_t determines the current output states$_t$, where b is the bias for each gate.

4.4.3 Gate Recurrent Unit Model

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks, established in 2014 [87]. Comparable to an LSTM with a forget gate, the GRU eliminates an output gate, resulting in fewer parameters on several tasks, like modeling polyphonic music and voice signals and NLP, GRU outperformed LSTM [88]. [89] [90] GRUs have been demonstrated to perform better on smaller and less frequent datasets [91].

Figure 4.5 shows a structural and schematic representation of GRU, which is an enhancement of the hidden layer in traditional RNNs. The three gates that make up a GRU are the update gate, the reset gate, and the temporary output. In this context, the following symbols are used:

1. x_t Is the network input at moment t.

2. h_t and $\overline{h_t}$ vectors of data representing the outputs of the temporary layer and the hidden layer at time t.

3. z_t and r_t are vectors that represent the state of the update and reset gates at time t, respectively, and are called gate outputs.

4.W_z,W_r, and W are corresponded, in order, to the weight matrices of the reset gate, the update gate, the temporary output, and the output layer;

5. b_r, b_z, and b_o are correlated to the biases that are associated with the weight matrices.

6. The sigmoid and tanh activation functions are represented by (X) and tanh (x), respectively.

Their exact phrases are as follows:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{4.8}$$

$$\frac{d\sigma}{dx} = \sigma(x)[1 - \sigma(x)] \tag{4.9}$$

$$\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)} \tag{4.10}$$

$$\frac{d\tanh}{dx}(x) = 1 - \tanh^2(x) \tag{4.11}$$

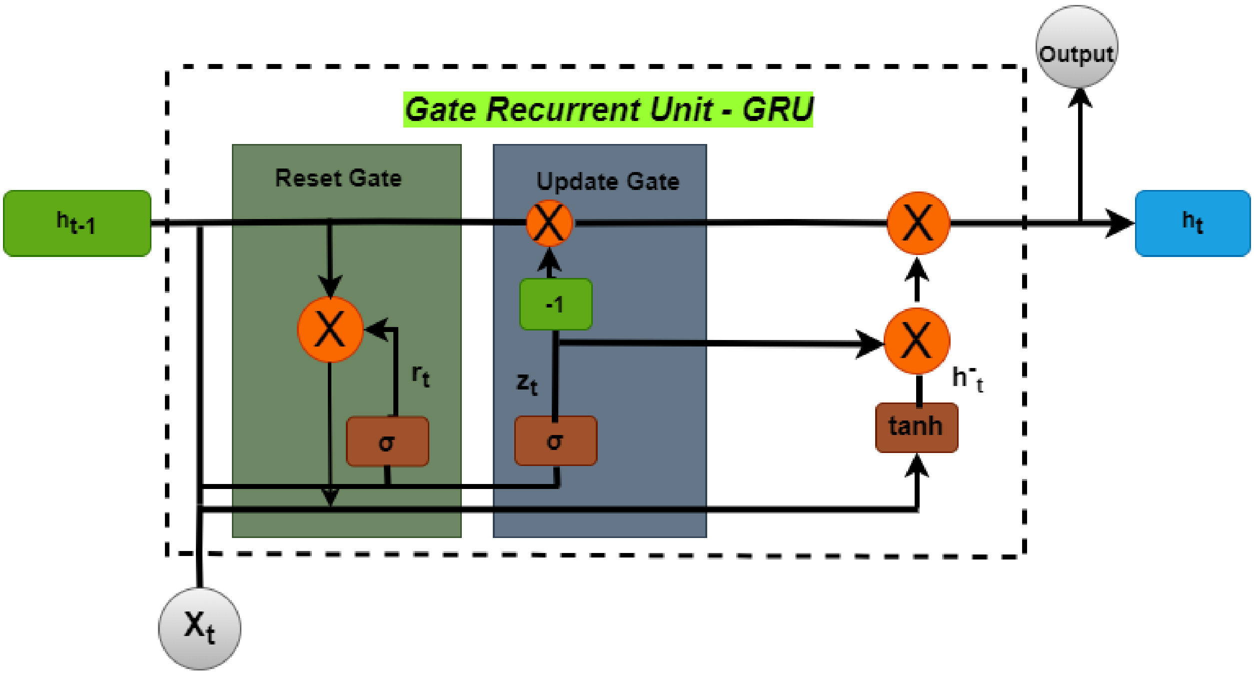

Figure 4.5: Gate recurrent unit structure.

GRU's mathematical model may be stated as follows: (4.12).

$$\begin{aligned} z_t &= \sigma_g(W_z + U_z h_{t-1} + b + z) \\ r_t &= \sigma_g(W_r x_t + U_r h_{t-1} + b + r) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot (w_h x_t + U_t(r_t \odot h_{t-1}) + b_h) \\ \overline{h_t} &= \tanh(W + r_t U h_{t-1} + b) \end{aligned} \tag{4.12}$$

Where $\odot$ is a multiplication sign, and x_t is an input variable that refers to load, and time in this case, h_t is an output variable that refers to forecasted residential load in this case, z_t is an updated gate variable, r_t is a reset gate, $\overline{h_t}$are a candidate memory cell, and W, U, and b are feature matrices.

The update gate is responsible for deciding how much historical data must be sent to the next level. This is particularly effective since it means the model may decide to recreate all historical data, wiping out any possibility of a decreasing gradient. For this approach, the reset gate is used to establish how much of the previous state of the cell may be disregarded. The first step is to flip the reset switch, at which point all data that was useful in the previous time step is copied into the newly allocated memory space. Weights are applied to both the input and the hidden layer, and the products are used. The concealed state multiple and the reset gate are then multiplied element by element. The subsequent sequence is produced by applying the non-linear activation function after the prior phases have been completed.

4.5 Hyperparameters Tuning For Machine Learning Models

This section will show the hyperparameters that were used in this research to get the best results for the models that have been applied. That is, will discover what are the best parameters that determine the structure of models to forecasting electrical loads, and it was called hyperparameter tuning.

Tuning is the problem of choosing an optimal set of hyperparameters for the learning algorithm [92]. The parameters will be as follows:

- Best optimizer.
- Activation function.
- Learning rate.
- Number of epochs.

- Batch size.
- Number of hidden layers.
- Dropout.

4.5.1 Optimizer

The optimizer was responsible for updating the weights and biases. This research applied more than one optimizer (Adam, Adagrad, RMSprop, AdaDelta) to find the best optimizer to give less error and high accuracy in forecasting short-term electrical loads.

4.5.1.1 Adam Deep Learning Optimizer

This optimization approach extends stochastic gradient descent to modify network weights during training. Adam optimizer changes the learning rate with each network weight independently [58].

The Adam optimizer provides various advantages that make it popular. It has been used as a benchmark for deep learning publications and is suggested as the default optimization approach. Furthermore, the method is easier to construct, has a shorter running time, consumes less memory, and needs fewer tuning compared to any other optimization technique. Large datasets benefit from this sort of optimizer. This optimizer is a hybrid of the Momentum and RMSP optimization techniques. This approach is really simple, easy to use, and takes minimal memory [93].

4.5.1.2 Adagrad (Adaptive Gradient Descent) Deep Learning Optimizer

AdaGrad is a sub-gradient technique series for stochastic optimization. Naturally, it adjusts the learning rate for each feature based on the predicted geometry of the issue; in particular, it assigns greater learning rates to rare features, ensuring that parameter changes are based on relevance rather than frequency. It is widely regarded as among the most commonly used algorithms (especially for training deep neural networks), and it spurred the creation of the Adam algorithm [94].

AdaGrad's goal is to minimize the estimated value of a stochastic optimal solution concerning a set of characteristics, based on a sequence of functional implementations. It does this, like other sub-gradient-based approaches, by modifying the variables in the reverse way of the sub-gradients. AdaGrad changes the learning rate for each parameter independently using the series of gradient forecasting, whereas typical sub-gradient algorithms utilize updating criteria with step sizes that disregard information from the previous.

4.5.1.3 Root Mean Square Propagation- RMSProp

RMSProp stands for the Robust Multi-Stage Progressive Optimization of NN. Since it is an extension of the SGD algorithm, the momentum methodology, and the foundation of the Adam technique, RMSProp has grown in prominence as an adaptive rate of learning approach over the years. Some examples of how RMSProp may be put into practice include the stochastic method for mini-batch gradient descent [95].

RMSProp is an optimized machine learning technique that uses multiple adaptive learning rates to train a neural network. It is inspired by the principles of gradient descent and RProp. RMSProp can achieve a quicker convergence rate than that of the basic optimizer, but less than sophisticated optimizers such as Adam, by combining averaging across mini-batches, efficiency, and gradients over subsequent mini-batches. Given the strong efficiency of RMSProp and the ability to integrate it with other algorithms, more difficult problems may be better characterized and converged within near.

4.5.1.4 Adadelta Deep Learning Optimizer

Adadelta is an improvement to the Adagrad machine learning optimization technique that addresses the method's two key flaws. Adagrad improves on prior gradient descent-based techniques by dynamically adjusting the learning rate (n) parameters for every scale in a system, allowing deep neural networks with thousands of dimensions to be trained in a manner that is neither excessively volatile and inaccurate nor too sluggish [96].

It solves the problem of a constantly decreasing learning rate throughout learning, which means that after several iterations, the learning rate will have become infinitesimally tiny. Earlier gradient descent techniques that required the determination of the global learning rate solved the problem. Adadelta was created to do away with the necessity for a learning rate. In this technique, recorded the square of gradients and refreshes in accumulators, but only in a limited way

4.5.2 Activation Function

The node's output is simply returned by this thing function. It's also known as the "transfer function" in certain contexts. It determines whether the answer given by a neural network is yes or no. The resulting values are converted between the ranges 0–1, -1–1, and so on (depending upon the function). You may use either a linear or a non-linear activation function, both of which are useful in different situations.

The nonlinear activation method functions are classified mostly based on their ranges or curves, as seen below:

1- Sigmoid or logistic activation function:

   Used a sigmoid function because of the transition (0 to 1). As a consequence, it works well for models where we need to foresee the likelihood as an output. The sigmoid distribution is optimal due to the limited range of probabilities (0, 1) that may be considered.

2- Tanh is also known as the hyperbolic tangent activation function.

3- Is a robust version of the logistic sigmoid. Within this interval, the tanh function is valid (-1 to 1). Even tanh has a sigmoid form (s-shaped).

4- Activation Function for ReLU (Rectified Linear Unit).

The ReLU is now the most popular activation function used across the world. It has now been used in almost all convolutional neural networks and deep learning methods. The monotonicity property holds for both the Leaky and Randomized ReLU functions. In addition, their derivatives vary continuously from zero to infinity and are monotonic.

4.5.3 Learning Rate

In machine learning and statistics, the learning rate is a tuning parameter for an optimization strategy that determines the mesh size at each iteration as it moves closer to the minimum loss function [97]. It is a metaphor for the rate at which a machine learning model "learns," since it indicates the extent to which freshly acquired information supersedes previously acquired wisdom. Adaptive control studies often refer to the learning rate as gain [98]. There is a compromise to be made between the convergence rate and overshooting when deciding on a learning rate. The learning rate represents the size of the step taken in the right direction, whereas the gradient of the loss function often determines the route of descent. If the learning rate is high enough, the process will skip over the minima, whereas if it's too slow, convergence will be slow or it will become stuck in a local minimum that's not desirable [99].

To improve quicker convergence, avoid oscillations, and avoid being stuck in unwanted local minima, the learning rate is frequently changed during training, either according to a learning rate schedule or by utilizing an adjustable learning rate [100]. The learning rate and its modifications may also vary depending on the parameter, resulting in a diagonal matrix that may be viewed as an approximation to the reverse of the Hessian matrix in Newton's approach [101]. The step length defined by imprecise line search in quasi-Newton techniques and similar optimization algorithms determines the learning rate [102]. Figure 4.6 shows the changes in the loss function vs. the epoch by the learning rate.

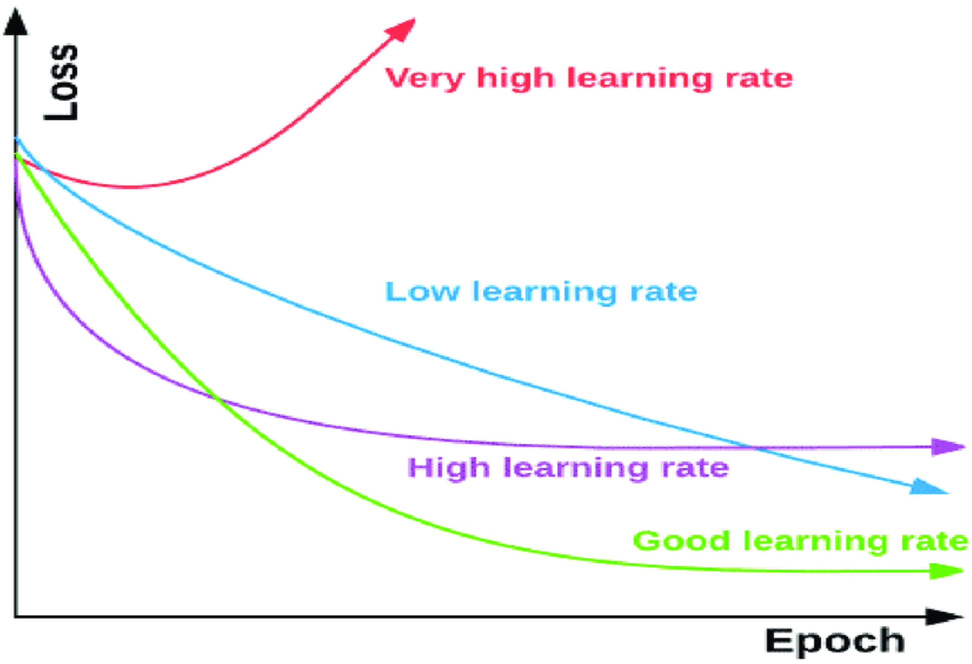

Figure 4.6: learning rate leads to a change in the loss function versus the epoch [103].

Figure 4.6 contrasts the learning rate-based changes in the loss function with the epoch. It takes a long time to identify the optimum answer when you have a poor learning rate. Furthermore, if the learning rate is excessively high, the best mode is rejected. Because a high learning rate is desirable in the initialization step and a lower one of these is advantageous in later stages, a slowing learning rate as the algorithm continues is preferable.

4.5.4 Epochs and Batch Size

An epoch is a term used in deep learning to describe the number of times an algorithm has been applied to the whole training dataset [104]. In most cases, data sets are broken up into groups called "batches" (especially when the amount of data is very large). Many people use the phrase iterations in a metaphorical sense, meaning that they want to run a single batch through the model several times. The number of epochs is equal to the number of iterations if the batch size is equal to the full training sample. For obvious reasons, this is almost never the case.

Batch size, such as the number of epochs, is a hyperparameter with no hard and fast rules. A small batch size introduces a significant degree of variation (noisiness) inside each batch since a tiny sample is unlikely to be a fair representation of the complete dataset. In contrast, if a batch size is too high, it could not fit in the buffer of the computer instances used for training and will overfit the dataset [105]. The batch size is controlled by other hyperparameters like learning rate, therefore the mixture of these hyperparameters is just as significant as the batch size itself.

4.5.5 Dropout

Dropout is a normalization technique that reduces overfitting in neural networks by preventing intricate co-adaptations on training data. It is an extremely effective method of averages models with neural networks [106]. Dropout refers to the elimination of cells (both hidden and apparent) from a neural network [107], [108].

A completely linked layer is prone to overfitting since it fills the majority of the features. A dropout is one way to reduce overfitting [107]. At each training iteration, a subset of nodes is "dropped out" of the network by removing them and their associated incoming and output edges. There, just the limited network receives data for training. Following this, the previously removed nodes are readmitted to the system with their original heft. Discarding a hidden node during training has a 0.5 probability; for input nodes, this should be much lower, presumably due to the loss of information.

4.5.6 Hidden Neural Network Layers

A hidden layer is a layer in an ANN that sits between the input and output layers and is where artificial neurons take in a batch of inputs and generate an output using an activation function.

It is a staple of the vast majority of artificial neural networks that attempt to simulate the workings of the human brain.

There is considerable leeway in how one sets up the hidden layers of a neural network. Weighted inputs are sometimes provided at random. In other circumstances, they are fine-tuned and validated via a technique known as backpropagation. In any case, the artificial neuron in the hidden layer functions similarly to a neuron in the brain, taking in random input signals, processing them, and converting them into an output matching the axons of the neuron. Several machine learning model assessments concentrate on the formation of hidden layers in the neural network. There are several methods to configure these hidden layers to produce diverse outputs, like DNN for machine vision, RNN with memory, and basic FNN, which function simply on training sets of data.

Based on the steps of the optimization and tuning process in the models, it shows the complete structural Figure 4.7 of building the model to get the best results.

4.6 Metrics Selection

Several metrics associated with data regression statistically measure its performance [109]. This thesis will focus on the following metrics for deep learning models, the next were used: MSE, RMSE, MAE, and R^2. This is to test these models and choose the best one based on the results to get the best forecasting of short-term electrical loads.

Mean Square Error (MSE) is a calculation of the mean squared deviation between observed and predicted values. Equation 4.13 shows how to calculate MSE.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (yt - \hat{y})^2 \qquad (4.13)$$

Root Mean Square Error (RMSE) is equal to the square root of the average squared error. Equation 4.14 shows how to calculate RMSE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (yt - \hat{y})^2} \qquad (4.14)$$

Mean Absolute Error (MAE) is the mean of the absolute value of the errors. Equation 4.15 shows how to calculate MAE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|y_{i-}\widehat{y_i}|) \qquad (4.15)$$

The coefficient of Determination (R^2) is a number between 0 and 1 that measures the accuracy with which a model can anticipate a given result. Equation 4.16 shows how to calculate R^2.

$$coefficient\ of\ determination(R^2) = 1 - \frac{SS_{regression}}{SS_{total}} \qquad (4.16)$$

Where:

- $SS_{regression}$ – The regression sum of squares (explained sum of squares)
- SS_{total} – The sum of all squares

4.7 Summary and Conclusion of the methodology

The first step of the methodology is data preparation and exploration, the second step is data preprocessing for machine learning; the third step is to use different machine learning algorithms for forecasting the electrical short-term load forecasting, and finally use of different performance metrics to compare different machines learning algorithms performance and select the best approach. Based on the steps of the optimization and tuning process in the models, it shows the complete structural Figure 4.7 of building the model to get the best results.

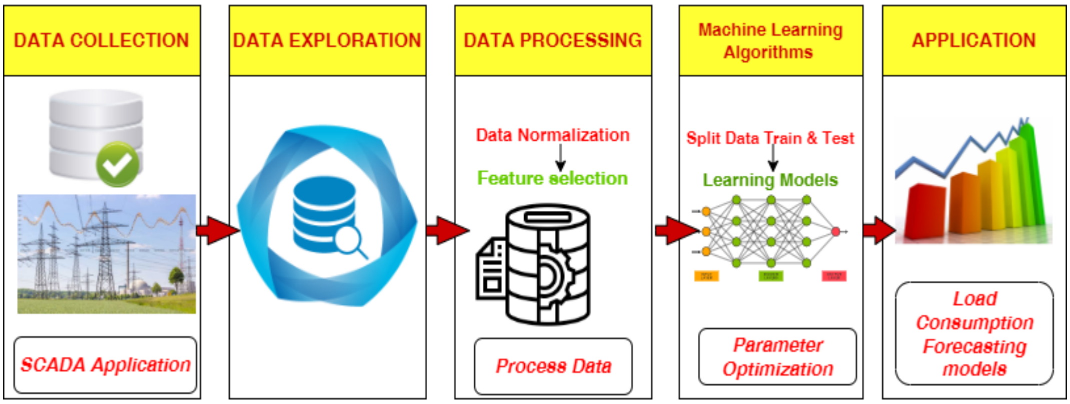

Figure 4.7: Basic workflow load consumption forecasting models.

Figure 4.7 shows the structural process, starting from data preparation, the training, and the testing process, in addition to the tuning that takes place in each model to choose the best and least error during the model training period. This is to obtain the best deep learning algorithm used to be finally relied upon.

It is also evident from the complete structural Figure 4.8 with the measures used in selecting the best algorithm in deep learning that has been used from the algorithms used to forecasting electrical loads.

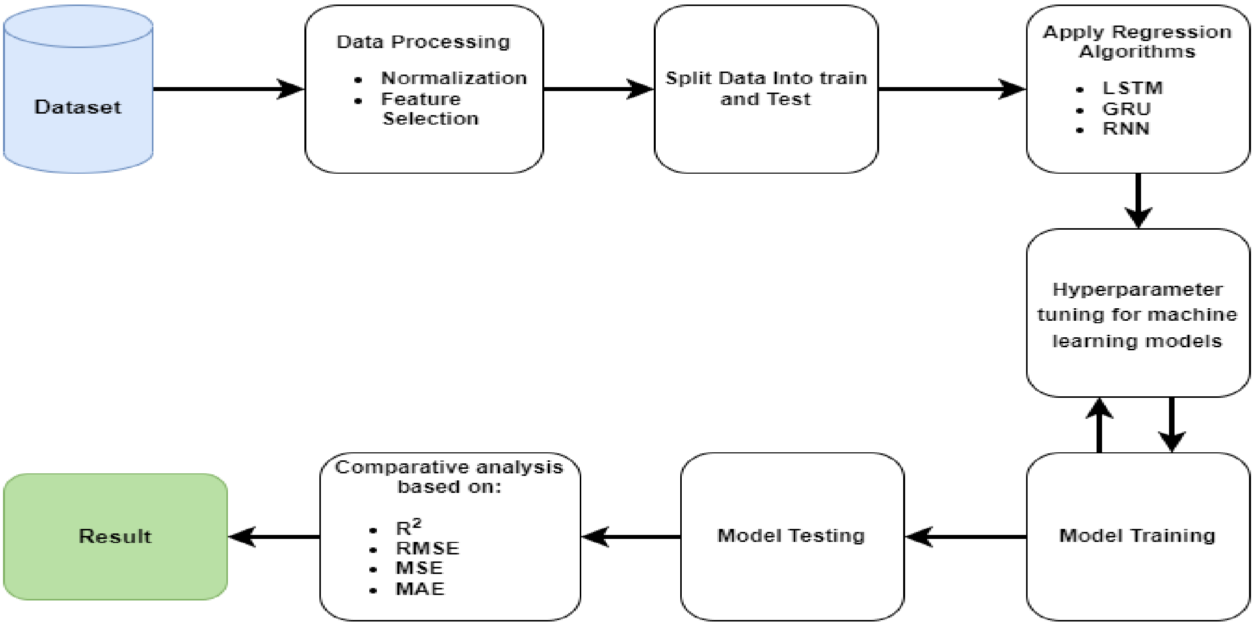

Figure 4.8: workflow Structure for Load Consumptions Forecasting.

Figure 4.8 shows the complete methodology used to green and build deep learning models. Where at the beginning the data is obtained and prepared in the best way, and then the operations are done on the data and divided into a training and testing phase, and after that the application of deep learning algorithms LSTM, RNN, GRU, and then the application of the tuning process for the models based on the comparative analysis (R square, RMSE, MSE, MAE).

This chapter summarizes the methodology of forecasting and Regression of the electrical load. Therefore, it is concluded from this chapter that the deep learning algorithms (LSTM, RNN, GRU) used in forecasting electrical loads are algorithms that deal with historical data, and each algorithm has a different feature from the others. As well as to obtain the best results in terms of accuracy and the lowest error rate, and apply the tuning process to choose the best parameter, and also every optimizer in the machine learning algorithms must be tried to choose the best in giving the highest results. The results, each method in which the tuning process was used, and the results of each optimizer used with deep learning algorithms will be discussed in the next chapter.

Chapter Five

Result and Discussion

5.1 Introduction

After deploying the models (LSTM, RNN, GRU), the metrics are used to evaluate the models' efficiency with different regression algorithms. In this chapter, will illustrate the results of the forecasting. As well as the performance metrics that rely on MSE, R^2, RMSE, and MAE, for each algorithm.

5.2 Forecasting Experiments

This section will explain the method used to get the best results in the forecasting process in each model that was used, and it was as follows: the best number of hidden layers, batch size, learning rate, type of optimizer, and type of activation function. In the beginning, the tree models (LSTM, RNN, and GRU) were applied to a training rate of 70% and a test of 30%, to more than one type of optimizer on each model with a different learning rate to get the best results. The following sections will discuss the results obtained from different types of optimizers having different numbers of hidden layers.

5.2.1 Adam Optimizer

We introduce Adam, an adaptive estimate of lower-order moments-based probabilistic optimization approach that uses a gradient-based first-order optimization strategy. The method is easily implemented, quick to compute, memory-light, insensitive to gradient rescaling, and effective for problems involving massive quantities of data and/or parameters. To iteratively adjust network weights based on training data, it may be used in lieu of the standard stochastic gradient descent method.

Adam optimizer was used in this research with the algorithms RNN, LSTM, and GRU to forecasting the demand for electricity to get the least error rate during forecasting. In this research, more than one hidden layer has been applied and compared between them to get the best result, because they are simply mathematical functions, each of which is designed to produce specific outputs for an intended result.

5.2.1.1 One Hidden Layer

In this section, one hidden layer was applied in addition to the input and output layer on each model with the Dropout to reduce overfitting and improve model performance. Table 5.1 shows the results for each model that was applied to one hidden layer with a different learning rate.

Table 5.1: Result Adam optimizer with one hidden layer.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	LSTM	0.04070	-0.8415	0.20174	0.16742
0.01		0.00282	0.87239	0.05310	0.03937
0.001		0.00400	0.81900	0.06324	0.04786
0.1	GRU	0.03337	-0.5100	0.18268	0.16819
0.01		0.00374	0.83063	0.06118	0.04731
0.001		0.00280	0.87323	0.05293	0.03790
0.1	RNN	0.06331	-1.8648	0.25163	0.21106
0.01		0.00295	0.86647	0.05432	0.04115
0.001		0.00307	0.86104	0.05541	0.04065

Table 5.1 shows the test results after applying each model based on the hyperparameter mentioned above. It can be seen that a different learning rate was used for each model because the learning rate is important when forming the neural network. While a slow learning rate might lead the process to stop, a fast one can cause the model to converge on a poor answer too rapidly. Since it represents the degree, to which the weights are modified throughout practice. So it must be chosen correctly. The results for the LSTM model were the worst results on the rate of learning equal to 0.1 because the percentage of gradation or descent was high

during learning, and the best results were on the rate of learning equal to 0.01 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.1 and the lowest error rate on the learning rate of 0.001. Almost the same was true for the RNN model, where the rate of learning equal to 0.1 was the worst error rate, and at a rate of 0.01, it was the lowest rate of error achieved by the model. All in all, the best model that got the lowest error rate is GRU with a rate of learning equal to 0.001.

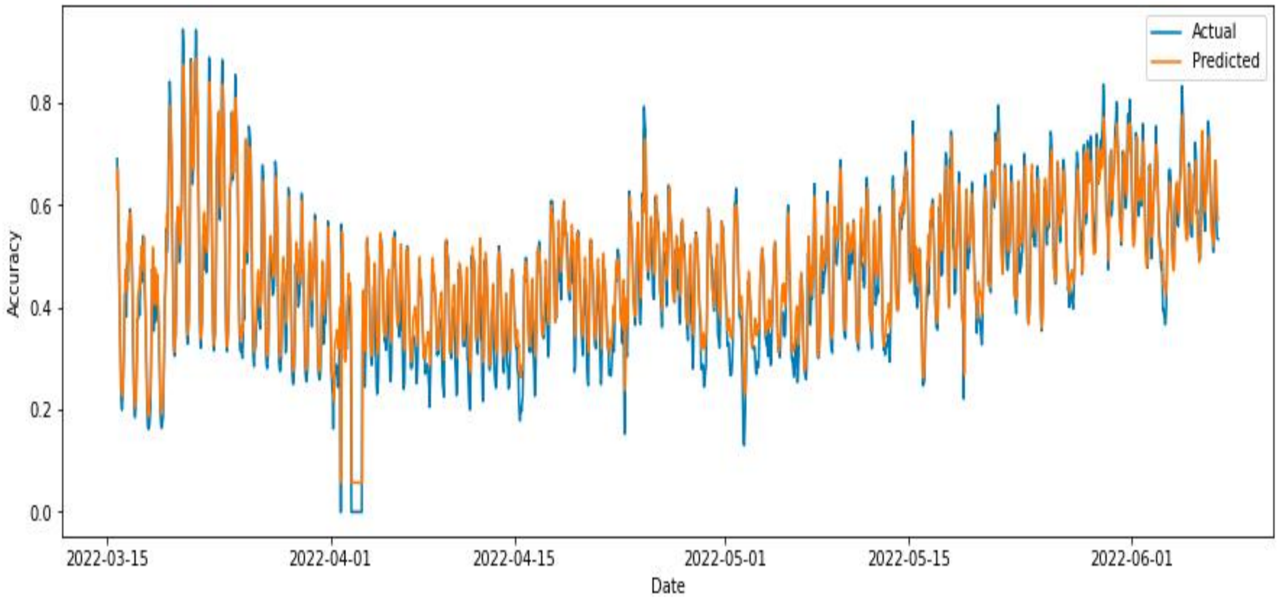

Figure 5.1: Long short-term memory result for forecasting electricity load value.

Figure 5.1 shows the difference between the actual values and the forecasting values to forecast electrical short-term loads of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the LSTM model, it was found that the LSTM model achieved the highest value at the learning rate of 0.01, where R^2 = 0.87239.

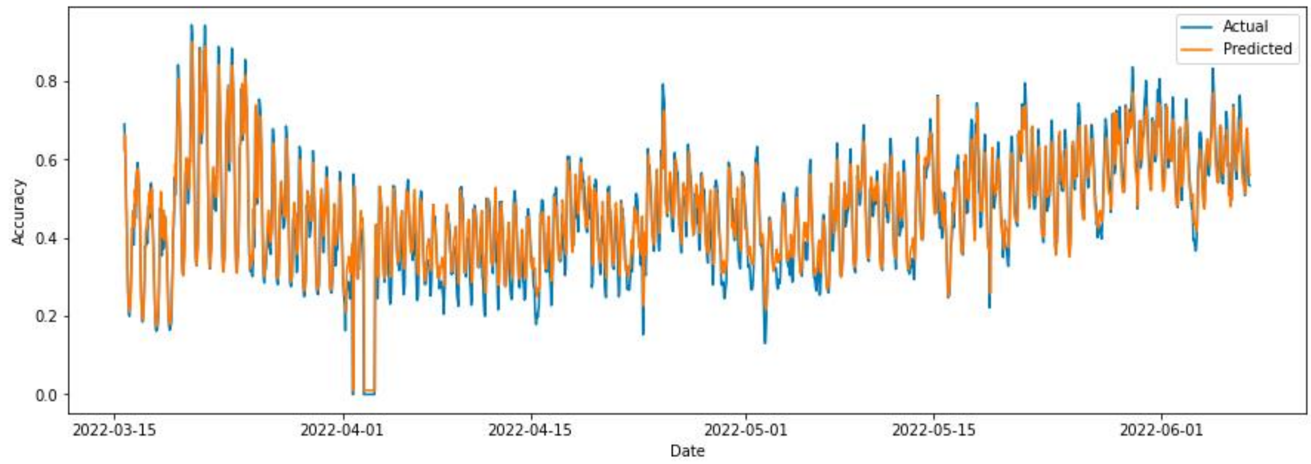

Figure 5.2: Gated recurrent unit result for forecasting electricity load value.

Figure 5.2 shows the difference between the actual values and the forecasting values to forecast electrical short-term loads of GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.001, where R^2 = 0.87323.

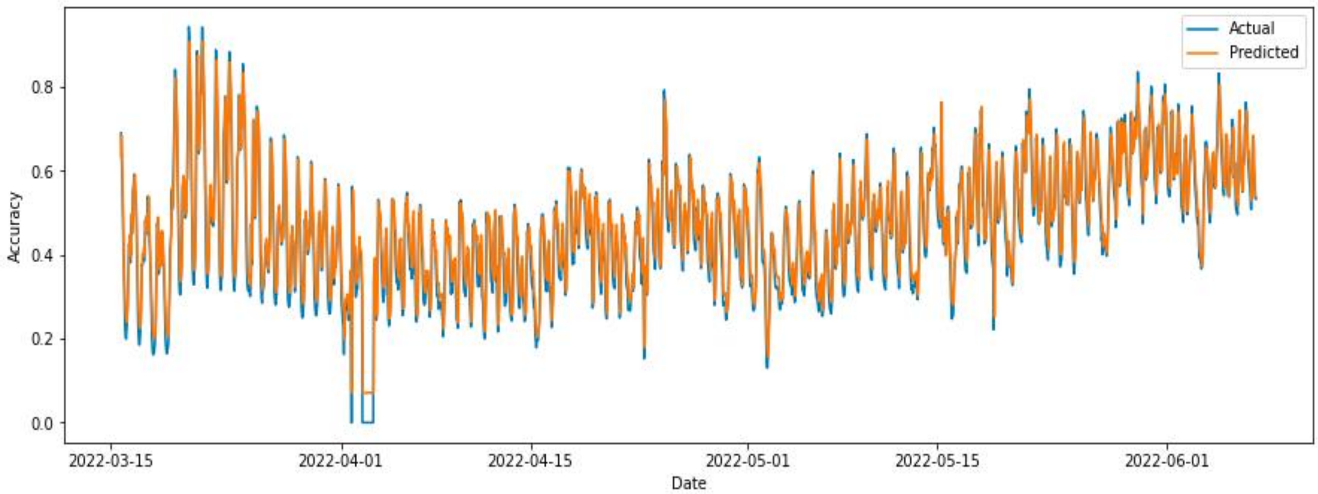

Figure 5.3: Recurrent neural network result for forecasting electricity load value.

Figure 5.3 shows the difference between the actual values and the predicted values to forecast electrical short-term loads of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the RNN model, it was found that the RNN model achieved the highest value at learning rate 0.01, where R^2 = 0.86647.

5.2.1.2 Two Hidden Layers

In this section, two hidden layers were applied in addition to the input and output layer on each model with the Dropout to reduce overfitting and improve model performance. Where In the input layer, its value is low = 0.2 because the model does not know all the data and in each hidden layer = 0.5. Table 5.2 shows the results for each model that was applied to two hidden layers and a different learning rate.

Table 5.2: Result Adam optimizer with two hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	LSTM	0.0404	0.8319	0.20122	0.1669
0.01		0.00293	0.8672	0.05417	0.04001
0.001		0.002988	0.864808	0.054662	0.04107
0.5		0.18453	-7.3492	0.42957	0.40334
0.05		0.00764	0.65425	0.08741	0.07211
0.005		0.00362	0.83583	0.06023	0.04584
0.1	GRU	0.0315	-0.427	0.1776	0.1451
0.01		0.00215	0.90228	0.04647	0.03266
0.001		0.0028	0.8727	0.0530	0.0384
0.5		0.724269	-31.7693	0.851039	0.837954
0.05		0.008508	0.615023	0.092242	0.079784
0.005		0.002806	0.873001	0.052980	0.038654
0.1	RNN	0.0355	-0.608	0.1885	0.1494
0.01		0.01529	0.30793	0.12367	0.10960
0.001		0.0038	0.8275	0.0617	0.0490
0.5		0.137202	-5.20766	0.370407	0.341961
0.05		0.193704	-7.76410	0.440118	0.414873
0.005		0.006286	0.7 15581	0.079285	0.064425

Table 5.2 shows the test results after applying each model based on the hyperparameter mentioned above. The first that a different learning rate was used for each model. Note that the results for the LSTM model were the worst results on the rate of learning equal to 0.5 because the percentage of gradation or descent was high during learning, and the best results were on the rate of learning equal to 0.01 where the error rate was the least possible. The results for the GRU model were having the worst error rate on the learning rate of 0.5 and the lowest error

rate on the learning rate of 0.01. Almost the same was true for the RNN model, where the rate of learning equal to 0.05 was the worst error rate, and at a rate of 0.001, it was the lowest rate of error achieved by the model. Overall, the best model that got the lowest error rate is GRU with a rate of learning equal to 0.01.

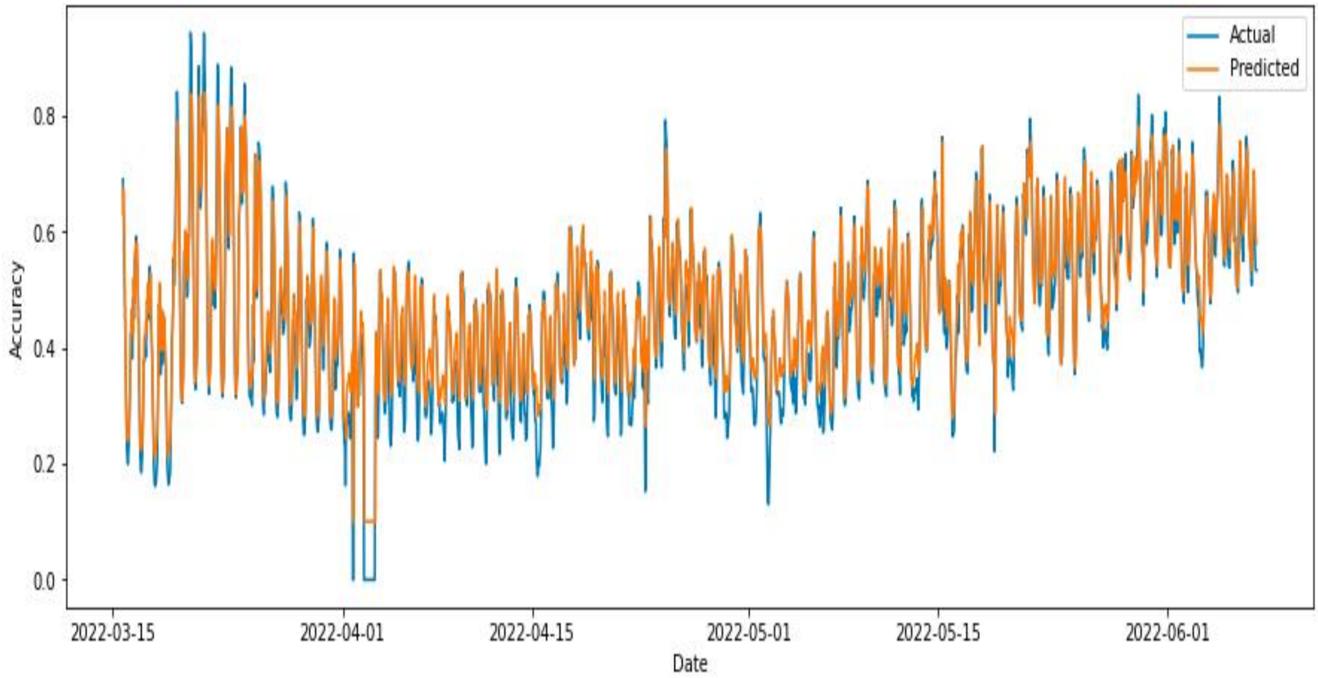

Figure 5.4 Long short-term memory result for forecasting electricity load value.

Figure 5.4 shows the difference between the actual values and the forecasted values to forecast electrical short-term loads of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the LSTM model, it was found that the LSTM model achieved the highest value at learning rate of 0.01, where $R^2 = 0.8672$.

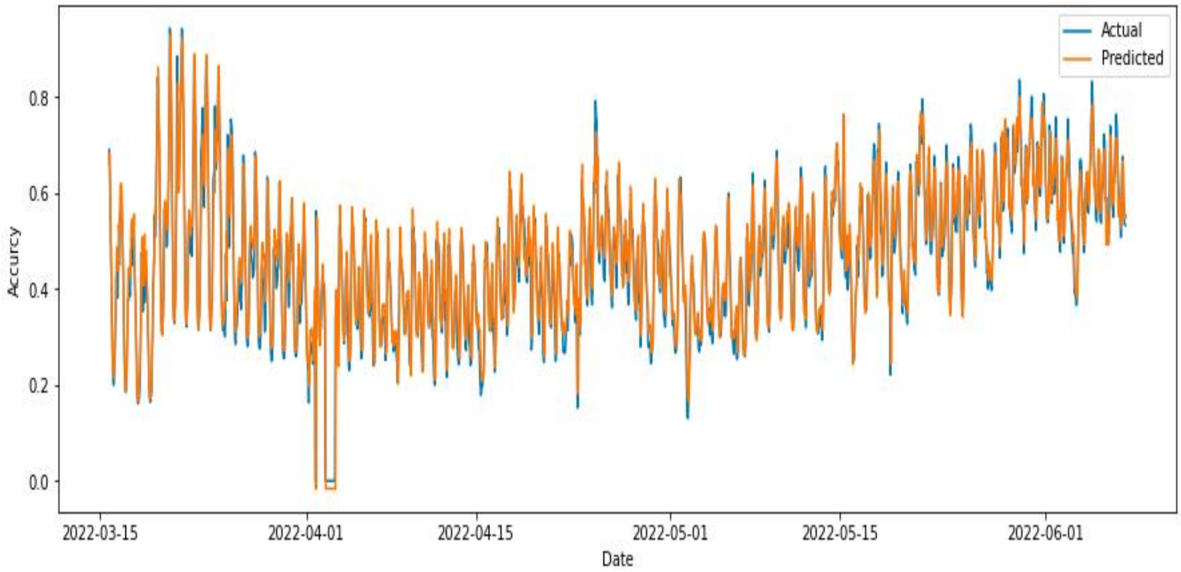

Figure 5.5 Gated recurrent unit result for forecasting electricity load value.

Figure 5.5 shows the difference between the actual values and the forecasted values to forecast electrical short-term loads of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the GRU model, it was found that the GRU model achieved the highest value at learning rate of 0.01, where R2 =0.90228

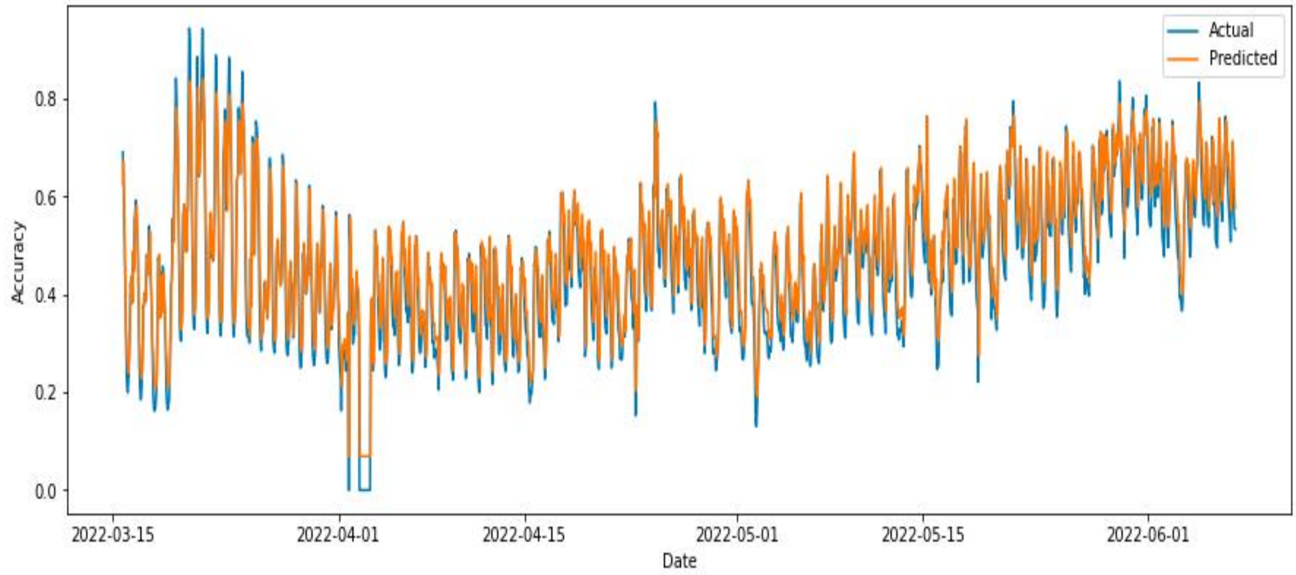

Figure 5.6 Recurrent Neural Network Result for Forecasting Electricity Load Value.

Figure 5.6 shows the difference between the actual values and the forecasted values to forecast electrical short-term loads of RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in RRN model, it was found that the RNN model achieved the highest value at learning rate 0.001, where R^2 = 0.8275.

5.2.1.3 Three Hidden Layers

In this section, three hidden layers were applied in addition to the input and output layers on each model with the Dropout. In the input layer, its value is low = 0.2 because the model does not know all the data, and each hidden layer = 0.5. Table 5.3 shows the results for each model applied to three hidden layers and a different learning rate.

Table 5.3: Result Adam optimizer with three hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	LSTM	0.04038	-0.8273	0.20096	0.16668
0.01		0.00378	0.82861	0.06154	0.04779
0.001		0.00312	0.85855	0.05591	0.04233
0.1	GRU	0.03154	-0.4270	0.17759	0.14508
0.01		0.00265	0.88001	0.05149	0.03738
0.001		0.00275	0.87547	0.05246	0.03790
0.1		0.04016	-0.8173	0.20041	0.16007

0.01	**RNN**	0.01614	0.26963	0.12705	0.10554
0.001		0.00432	0.80448	0.06573	0.05348

Table 5.3 shows the test results after applying each model based on the hyperparameter mentioned above. Different learning rate was used for each model. The results for the LSTM model were the worst on a rate of learning equal to 0.1 because the percentage of gradation or descent was high during learning, and the best results were on a rate of learning equal to 0.001 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.1 and the lowest error rate on the learning rate of 0.01. Almost the same was true for the RNN model, where the rate of learning equal to 0.1 was the worst error rate, and at a rate of 0.001, it was the lowest rate of error achieved by the model. As noted, the best model that got the lowest error rate is GRU with a rate of learning equal to 0.01.

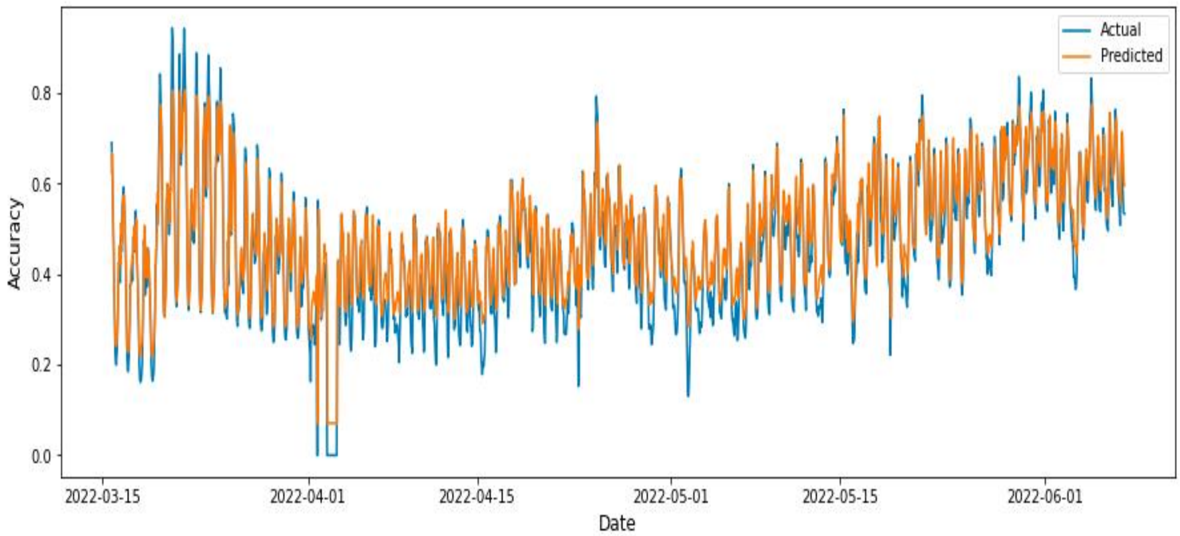

Figure 5.7: Long short-term memory result for forecasting electricity load value.

Figure 5.7 shows the discrepancy between the actual and forecasted results to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.001, where R^2 = 0.85855. As can be seen from Figure 5.7 the higher (peak) values of the model, cannot be forecasted accurately while applying three hidden layers.

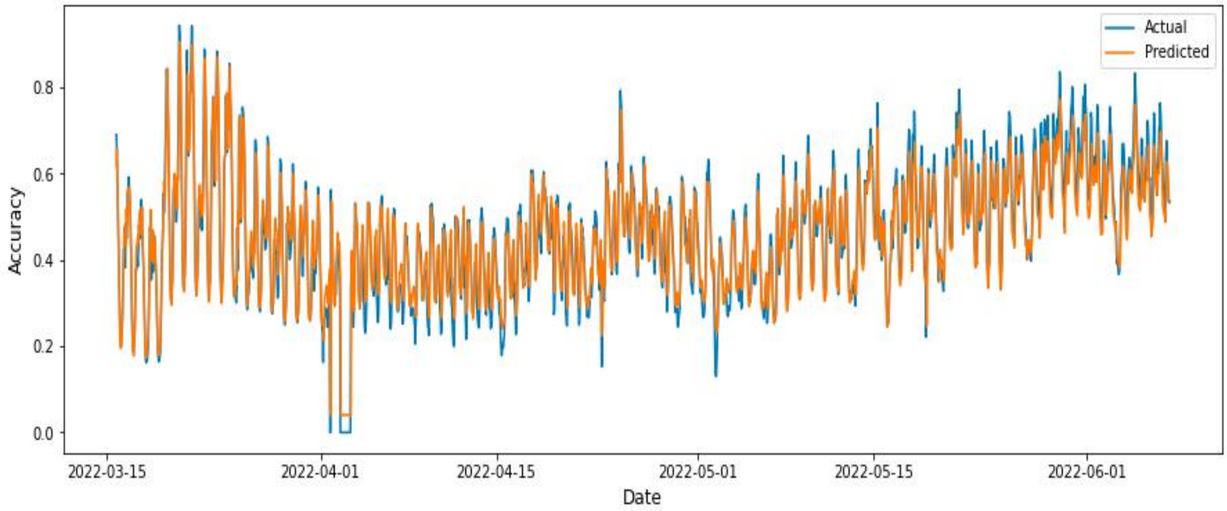

Figure 5.8: Gated recurrent unit result for forecasting electricity load value.

Figure 5.8 shows the discrepancy between the actual and forecasted results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.01, where $R^2 = 0.88001$. It can be seen from Figure 5.8 the higher (peak) values of the model, in this case, can forecast, but with lower accuracy, especially in the last forecasting test period, and the model was able to give better results than the LSTM model.

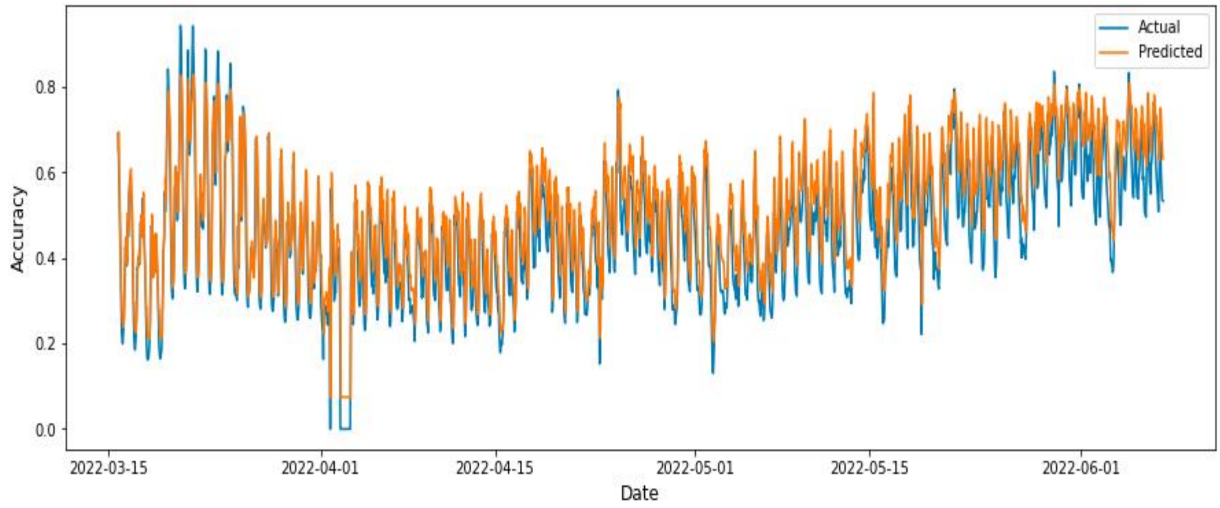

Figure 5.9: Recurrent neural network result for forecasting electricity load value.

Figure 5.9 shows the discrepancy between the actual and forecasting results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in RNN model, it was found that the RNN model achieved the highest value at learning rate 0.001, where R^2 = 0.80448. It can be seen from Figure 5.9 the model's higher values (peak) and lower values; in this case, the model was unable to forecasted the desired shape. The accuracy results were unsatisfactory because the model was able to forecast almost intermediate values and found it difficult to forecast the values at the top and the lower values. This model was the worst in forecasting values compared to other models.

After applying the Adam optimizer in more than one way (one hidden layer, two hidden layers, and three hidden layers) with machine learning models LSTM, RNN, and GRU. The results obtained from this optimizer were applied with two hidden layers With the GRU model, gave the best results, as the R^2 was 90.228%, RMSE is 0.04647, and the MAE was 0.03266.

5.2.2 AdaGrad Optimizer

In this section, the results will discuss after applying the AdaGrad optimizer. The goal of AdaGrad in this research is to reduce the expected value of a stochastic target function, concerning a set of parameters, given the sequence of job realization, and adjust the learning rate for each parameter individually using a sequence of gradient estimates. Based on the objectives of this optimizer, its application in forecasting electrical load in the short term helps to get the best results and the least error rate.

AdaGrade optimizer was used in this research with the RNN, LSTM, and GRU models to forecast the demand for electricity to get the least error rate during forecasting. In this section, more than one hidden layer has been applied and compared between them to get the best result, because they are simply mathematical functions, each of which was designed to produce specific outputs for an intended result.

5.2.2.1 One Hidden Layer

In this section, one hidden layer was applied in addition to the input and output layer on each model with the Dropout in AdaGrade optimizer. Table 5.4 shows the results for each model that was applied to one hidden layer and a different learning rate.

Table 5.4: Result AdaGrad optimizer with one hidden layer.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	**LSTM**	0.00295	0.86627	0.05436	0.04305
0.01		0.00822	0.62783	0.09069	0.07237
0.001		0.01718	0.22241	0.13109	0.10541
0.1	**GRU**	0.00319	0.85533	0.05654	0.04119
0.01		0.00300	0.86413	0.05479	0.04042
0.001		0.01129	0.48914	0.10625	0.08688
0.1	**RNN**	0.00303	0.86273	0.05508	0.04251
0.01		0.00320	0.86399	0.05489	0.04171
0.001		0.00353	0.84014	0.05944	0.04606

Table 5.4 shows the test results after applying each model based on the hyperparameter mentioned above. A different learning rate was used for each model. The results for the LSTM model were the worst results on the rate of learning equal to 0.001, and the best results were on rate of learning equal to 0.1 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.001 and the lowest error rate on the learning rate of 0.01. Almost the same was true for the RNN model, where the rate of learning equal to 0.001 was the worst error rate, and at a rate of 0.01, it was the lowest rate of error achieved by the model. All in all, the best model that got the lowest error rate is LSTM with a rate of learning equal to 0.1.

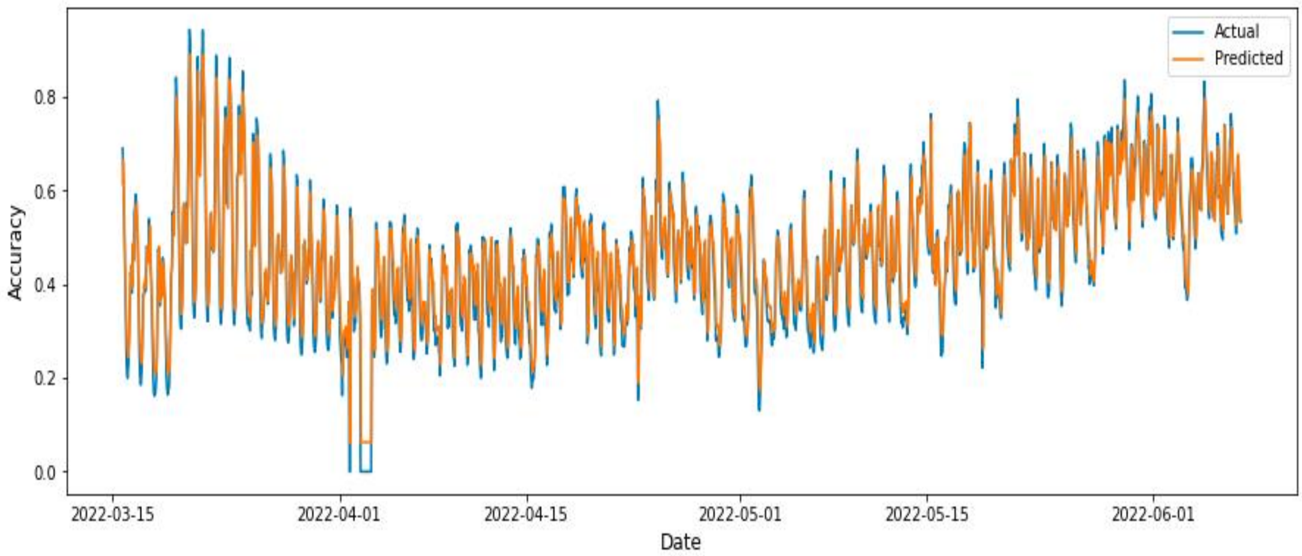

Figure 5.10: Long short-term memory result for forecasting electricity load value.

Figure 5.10 shows the discrepancy between the actual and predictive results to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.1, where $R^2 = 0.86627$. It can be seen from Figure 5.10 the error rate in forecasting the loads on the peak is very small.

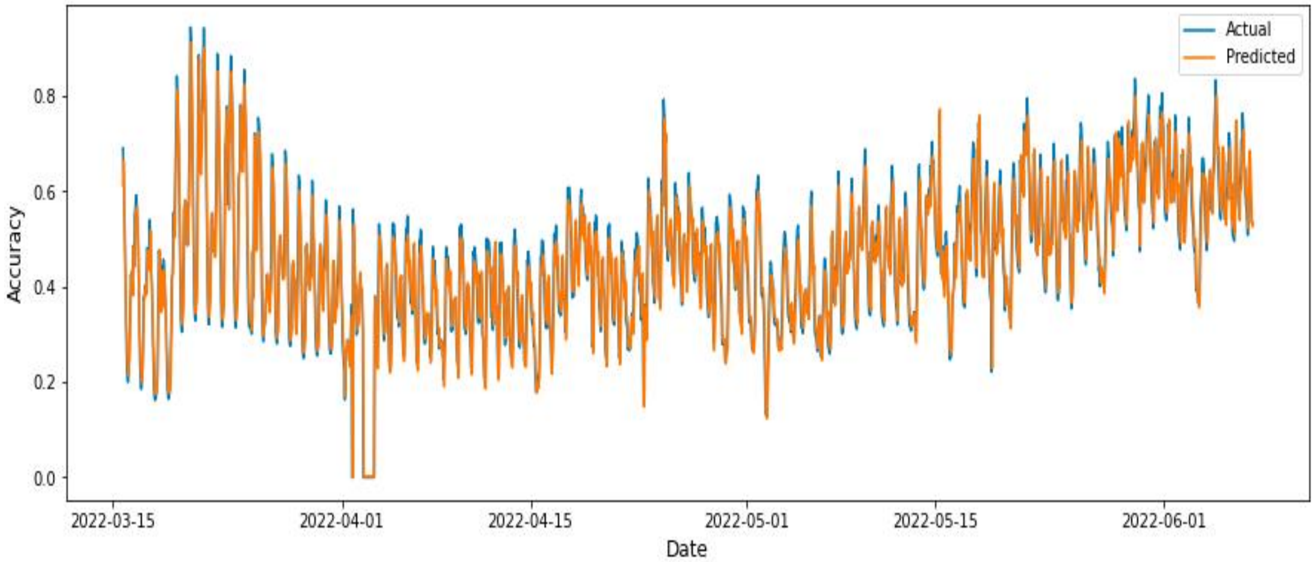

Figure 5.11: Gated recurrent unit result for forecasting electricity load value.

Figure 5.11 shows the discrepancy between the actual and forecasted results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.01, where R^2 = 0.86413. It can be seen from Figure 5.11, the error rate in forecasting the loads on the peak is very small, and the results are very close to the LSTM model.

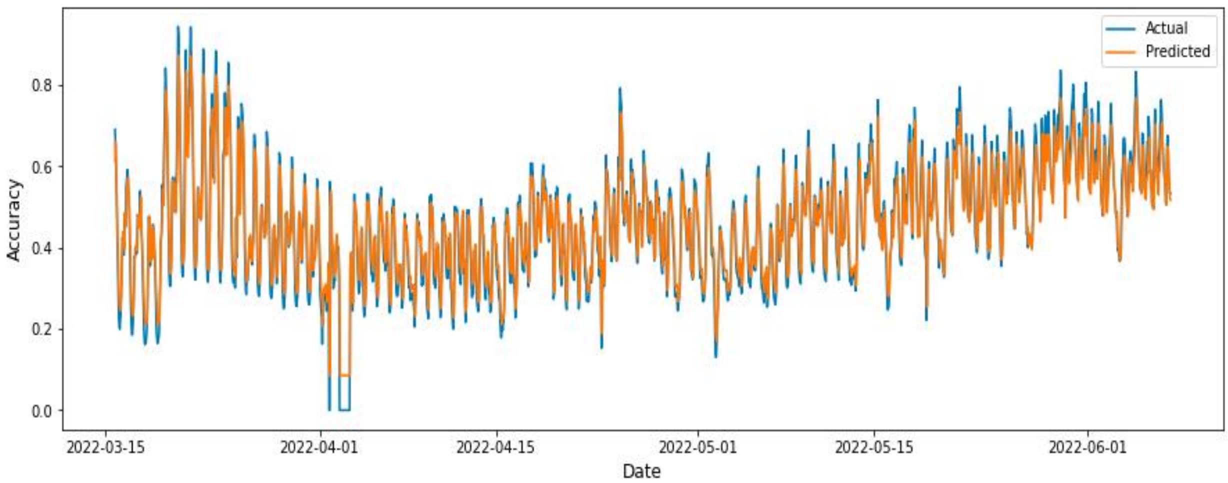

Figure 5.12: Recurrent neural network result for forecasting electricity load value.

Figure 5.12 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate.

After comparing the test results for each learning rate in RNN model, it was found that the RNN model achieved the highest value at learning rate 0.01, where $R^2 = 0.86399$. It can be seen from Figure 5.12, the error rate in forecasting the loads on the peak is very small, and the results are very close to the LSTM and GRU models.

5.2.2.2 Two Hidden Layers

In this section, two hidden layers were applied in addition to the input and output layer on each model with the Dropout to reduce overfitting and improve model performance. Table 5.5 shows the results for each model that was applied to two hidden layers and a different learning rate.

Table 5.5: Result AdaGrad optimizer with two hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	LSTM	0.0030	0.8600	0.0556	0.0436
0.01		0.0215	0.0263	0.1466	0.1171
0.001		0.0237	0.076	0.1542	0.1217
0.1	GRU	0.0035	0.8378	0.0598	0.0444
0.01		0.0029	0.8672	0.0541	0.0399
0.001		0.0131	0.4036	0.1148	0.0925
0.1	RNN	0.0040	0.8148	0.0639	0.0522
0.01		0.0031	0.8587	0.0558	0.0429
0.001		0.0063	0.7140	0.0794	0.0611

Table 5.5 shows the test results after applying each model based on the hyperparameter mentioned above. Different learning rate was used for each model. The results for the LSTM model were the worst results on rate of learning equal to 0.001, and the best results were on rate of learning equal to 0.1 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.001 and the lowest error rate on the learning rate of 0.01. Almost the same was true for the RNN model, where rate of learning equal to 0.001 was the worst error rate, and at a rate of 0.01, it was the lowest rate of error achieved by the model. All in all, the best model that got the lowest error rate is GRU with rate of learning equal to 0.01.

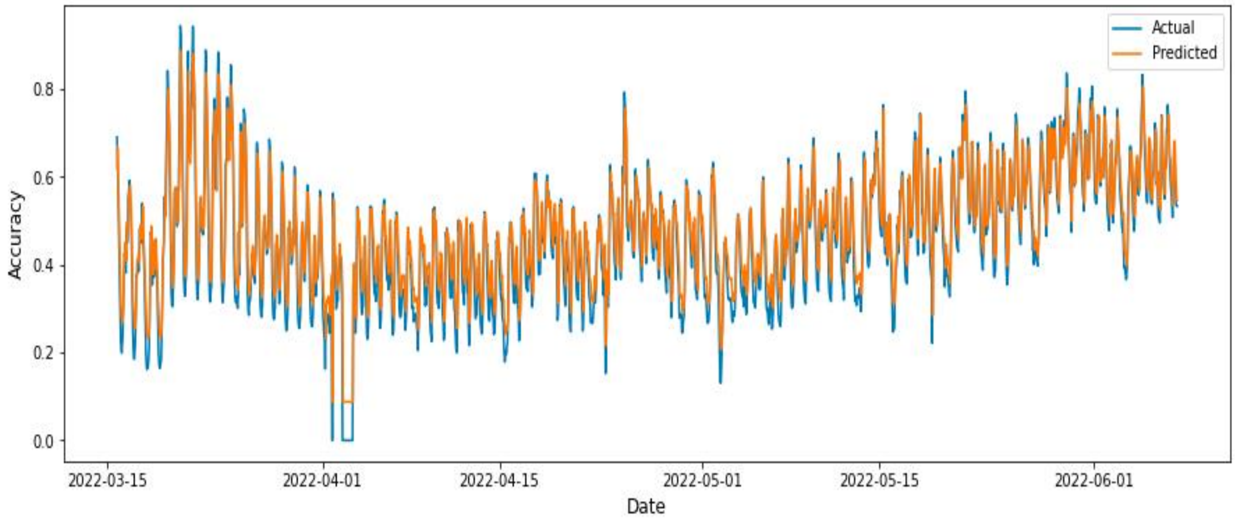

Figure 5.13: Long short-term memory result for forecasting electricity load value.

Figure 5.13 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.1, where $R^2 = 0.8600$. It can be seen from Figure 5.13, the error rate in forecasting the loads on the peak is very small, but there is a clear difference between the actual and forecasted values at the bottom (the electricity demand is minimal) in this model.

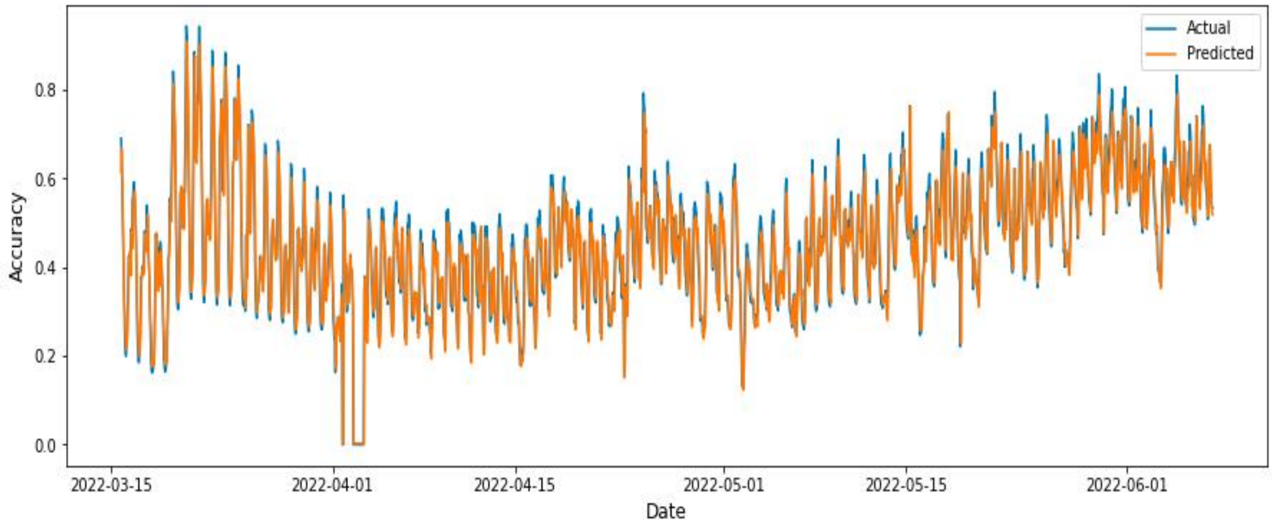

Figure 5.14: Gated recurrent unit result for forecasting electricity load value.

Figure 5.14 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the GRU model, it was found that the GRU model achieved the highest value at learning rate 0.01, where R2 = 0.8672. It can be seen from Figure 5.14, the error rate in forecasting the loads on the peak is very small.

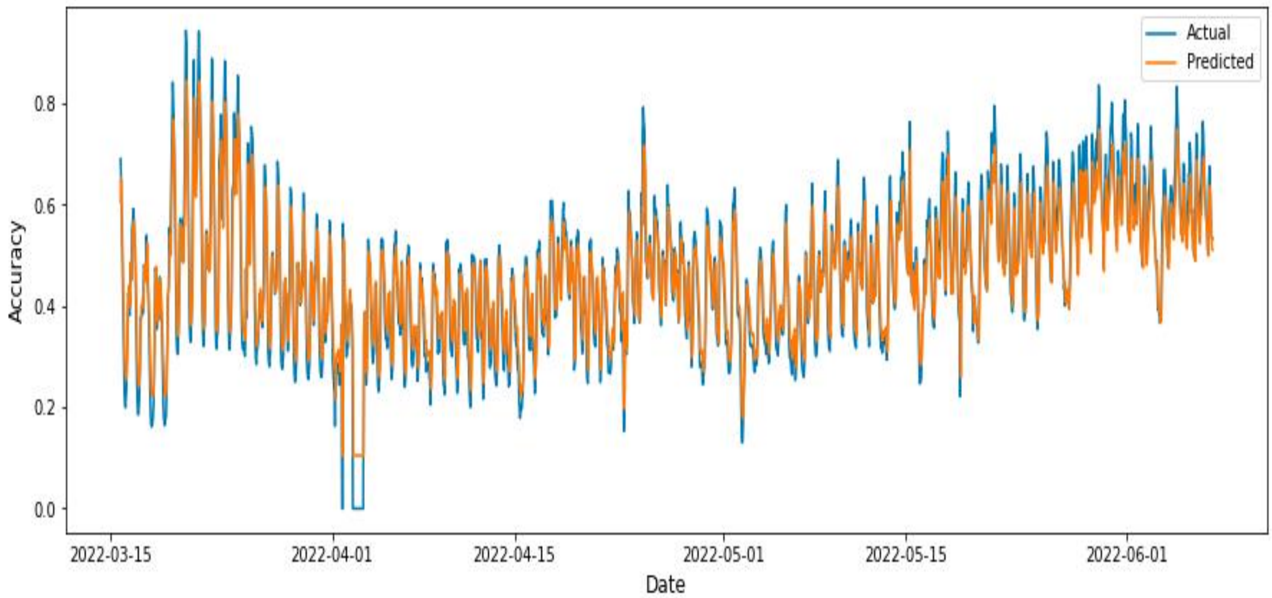

Figure 5.15: Recurrent neural network result for forecasting electricity load value.

Figure 5.15 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate.

After comparing the test results for each learning rate in RNN model, it was found that the RNN model achieved the highest value at learning rate 0.01, where R2 = 0.8587. It can be seen from Figure 5.15, is a clear difference between the actual and forecasted values at the bottom (the electricity demand is minimal) in this model, and in the same case in the peak value their a clear difference.

5.2.2.3 Three Hidden Layers

In this section, three hidden layers were applied in addition to the input and output layers on each model with the Dropout. Table 5.6 shows the results for each model applied to three hidden layers and a different learning rate.

Table 5.6: Result AdaGrad optimizer with three hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	LSTM	0.02191	0.00837	0.14804	0.11889
0.01		0.02224	-0.0066	0.14916	0.11908
0.001		0.02413	-0.0921	0.15536	0.12279
0.1	GRU	0.00405	0.81659	0.06366	0.04953
0.01		0.00301	0.86373	0.05487	0.04083
0.001		0.02001	0.09425	0.14148	0.11407
0.1	RNN	0.00908	0.58907	0.09530	0.08068
0.01		0.00329	0.85094	0.05739	0.04482
0.001		0.01153	0.47814	0.10739	0.08221

Table 5.6 shows the test results after applying each model based on the hyperparameter mentioned above. A different learning rate was used for each model. The results for the LSTM model were the worst results on rate of learning equal to 0.001, and the best results were on rate of learning equal to 0.1 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.001 and the lowest error rate on the learning rate of 0.01. Almost the same was true for the RNN model, where the rate of learning equal to 0.001 was the worst error rate, and at a rate of 0.01, it was the lowest rate of error achieved by the model. All in all, the best model that got the lowest error rate is GRU with rate

of learning equal to 0.01, and note that the results of the LSTM model are very few of accuracy with a very high error rate.

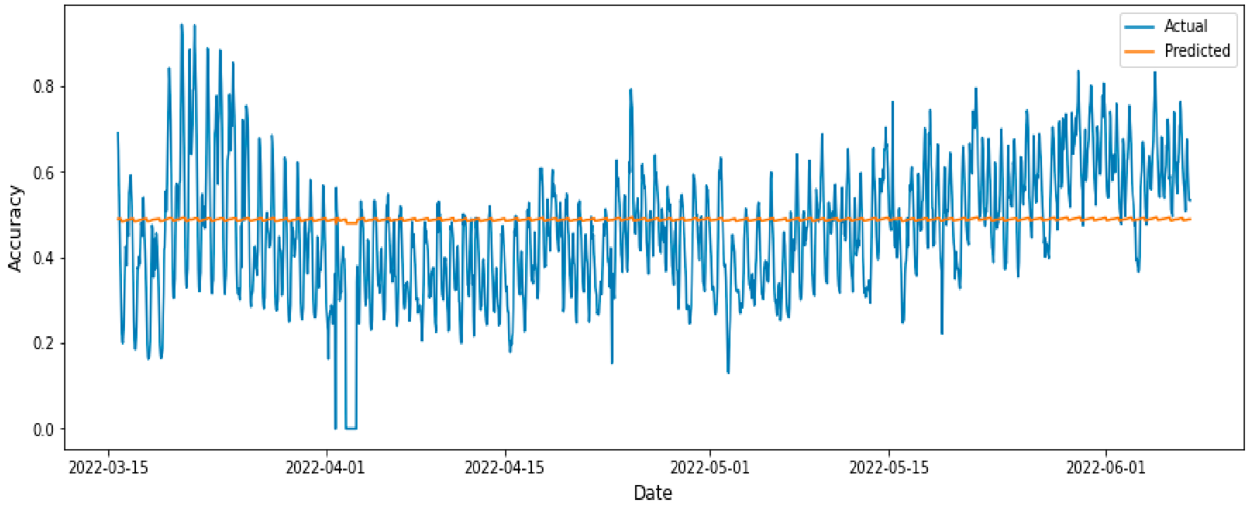

Figure 5.16: Long short-term memory result for forecasting electricity load value.

Figure 5.16 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. It can be seen from Figure 5.16, the LSTM model almost could not forecasted the electrical load when using this optimizer with three hidden layers, the error rate was very high and the difference between the real and forecast values is almost equal to the real values. This is an indication that using this enhancer with the LSTM model with three hidden layers cannot be applied to our study. In this case, the model could not predict, because the learning step was 0.1 and the model was unable to reach the global minimum of data to learn the model from all data.

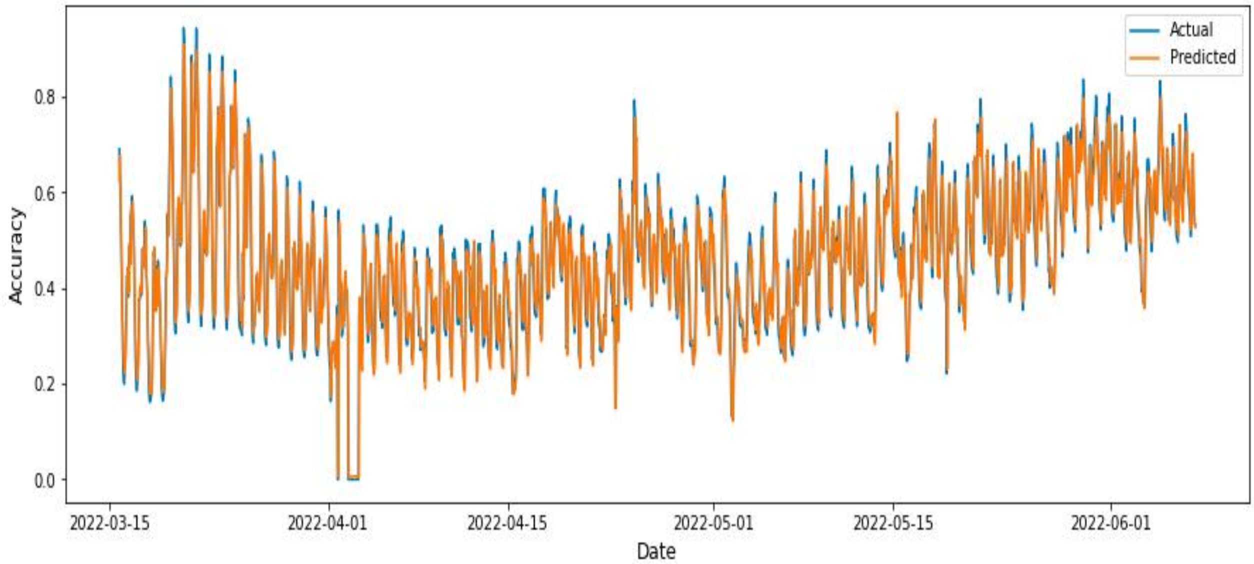

Figure 5.17: Gated recurrent unit result for forecasting electricity load value.

Figure 5.17 shows the discrepancy between the actual and forecasted results to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the GRU model, it was found that the GRU model achieved the highest value at learning rate of 0.01, where $R^2 = 0.86373$. It can be seen from Figure 5.17, the error rate in forecasting the loads on the peak is small.

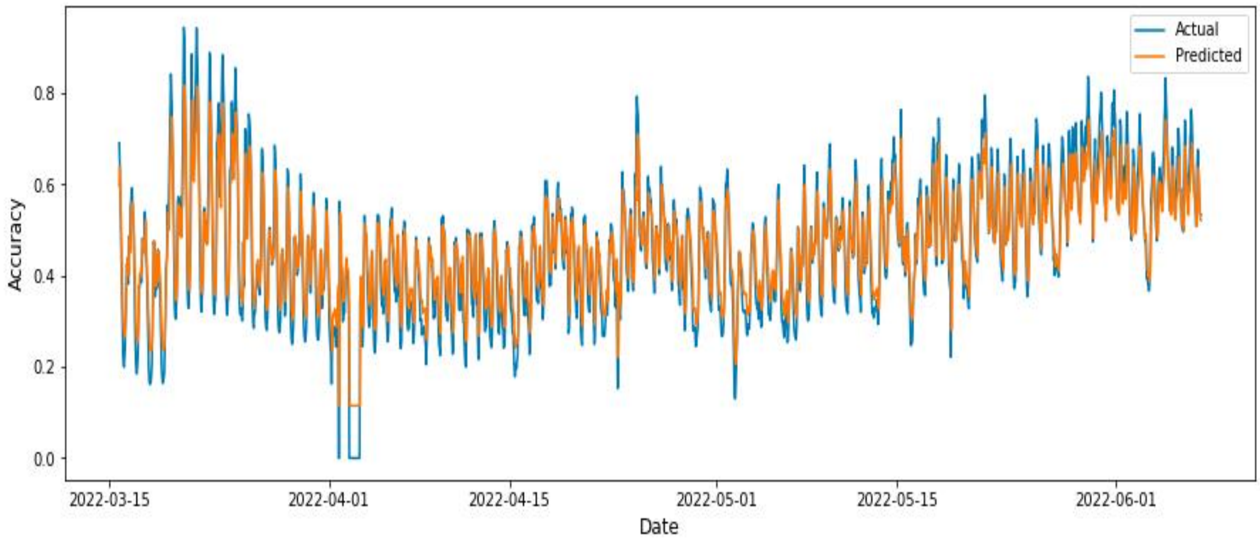

Figure 5.18: Recurrent Neural Network Result for Forecasting Electricity Load Value.

Figure 5.18 shows the discrepancy between the actual and forecasted results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate.

After comparing the test results for each learning rate in RNN model, it was found that the RNN model achieved the highest value at learning rate 0.01, where R2 = 0.85094. It can be seen from Figure 5.18, is a clear difference between the actual and forecasted values at the bottom (the electricity demand is minimal) in this model, and the same case in the peak value them a clear difference.

After applying the AdaGrad optimizer in more than one way (one hidden layer, two hidden layers, three hidden layers, different learning rates) with machine learning models LSTM, RNN, and GRU. The results obtained from this optimizer are applied with two hidden layers With the GRU model and the learning rate is 0.01, it gave the best results, as the R^2 was 86.72%, RMSE is 0.0541, and the MAE was 0.0399.

5.2.3 RMSprop Optimizer

In this section, the results will discuss after applying the RMSprop optimizer. The goal of RMSprop in this research is to train the ANN by different adaptive learning rates and is derived from the concepts of gradient descent and RProp. Based on the objectives of this optimizer, its application in forecasting electrical load in the short term helps to get the best results and the least error rate.

RMSprop optimizer was used in this research with the RNN, LSTM, and GRU models to forecast the demand for electricity to get the least error rate during forecasting. In this section, more than one hidden layer has been applied and compared between them to get the best result, because they are simply mathematical functions, each of which was designed to produce specific outputs for an intended result.

5.2.3.1 One Hidden Layer

In this section, one hidden layer was applied in addition to the input and output layer on each model with the Dropout in RMSprop optimizer. Table 5.7 shows the results for each model that was applied to one hidden layer and a different learning rate.

Table 5.7: Result RMSprop optimizer with one hidden layer.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	**LSTM**	0.00782	0.64599	0.08845	0.07215
0.01		0.00349	0.84209	0.05907	0.04313
0.001		0.00350	0.84130	0.05922	0.04489
0.1	**GRU**	0.02235	-0.0112	0.14950	0.11889
0.01		0.00270	0.87749	0.05203	0.03904
0.001		0.00354	0.83976	0.05951	0.04310
0.1	**RNN**	0.23414	-9.5938	0.48388	0.43031
0.01		0.00329	0.85114	0.05735	0.04446
0.001		0.00335	0.84833	0.05789	0.04609

Table 5.7 shows the test results after applying each model based on the hyperparameter mentioned above. Different learning rate was used for each model. The results for the LSTM model were the worst results on rate of learning equal to 0.1, and the best results were on rate of learning equal to 0.01 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.1 and the lowest error rate on the learning rate of 0.01. Almost the same was true for the RNN model, where rate of learning equal to 0.1 was the worst error rate, and at a rate of 0.01, it was the lowest rate of error achieved by the model. Overall, the best model that got the lowest error rate is GRU with rate of learning equal to 0.01.

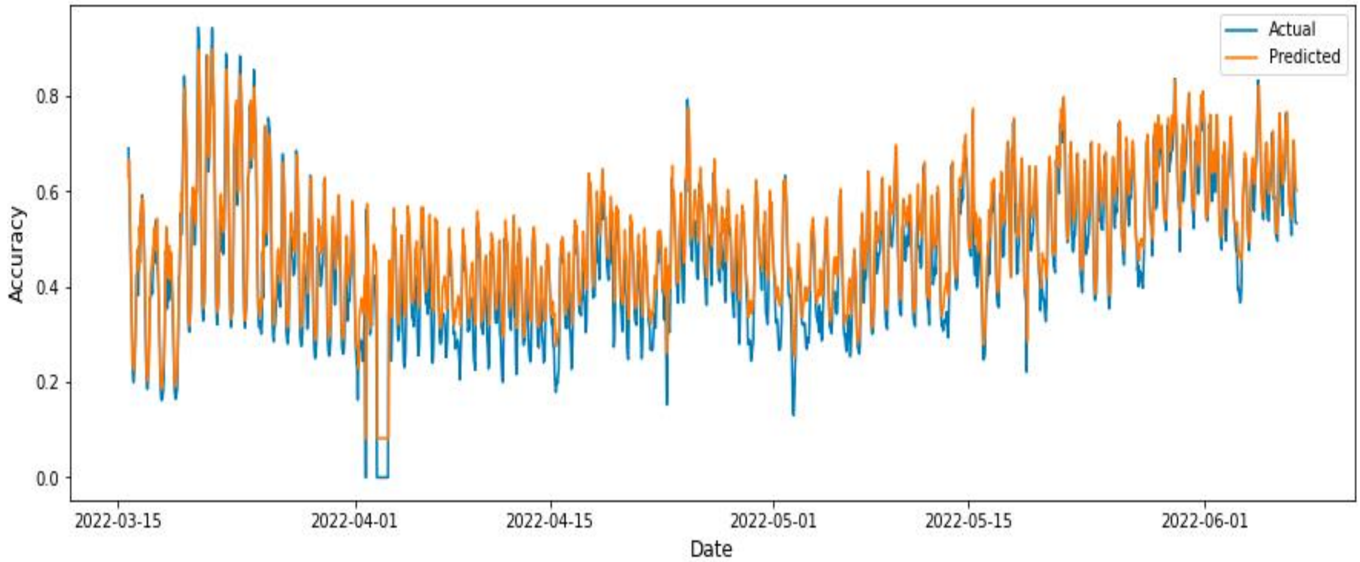

Figure 5.19: Long short-term memory result for forecasting electricity load value.

Figure 5.19 shows the discrepancy between the actual and forecasted results in order to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.01, where $R^2 = 0.84209$. It can be seen from Figure 5.19, there is an error rate in the difference between the real and forecasted loads, where the accuracy rate in the small loads (minimum load) is not high.

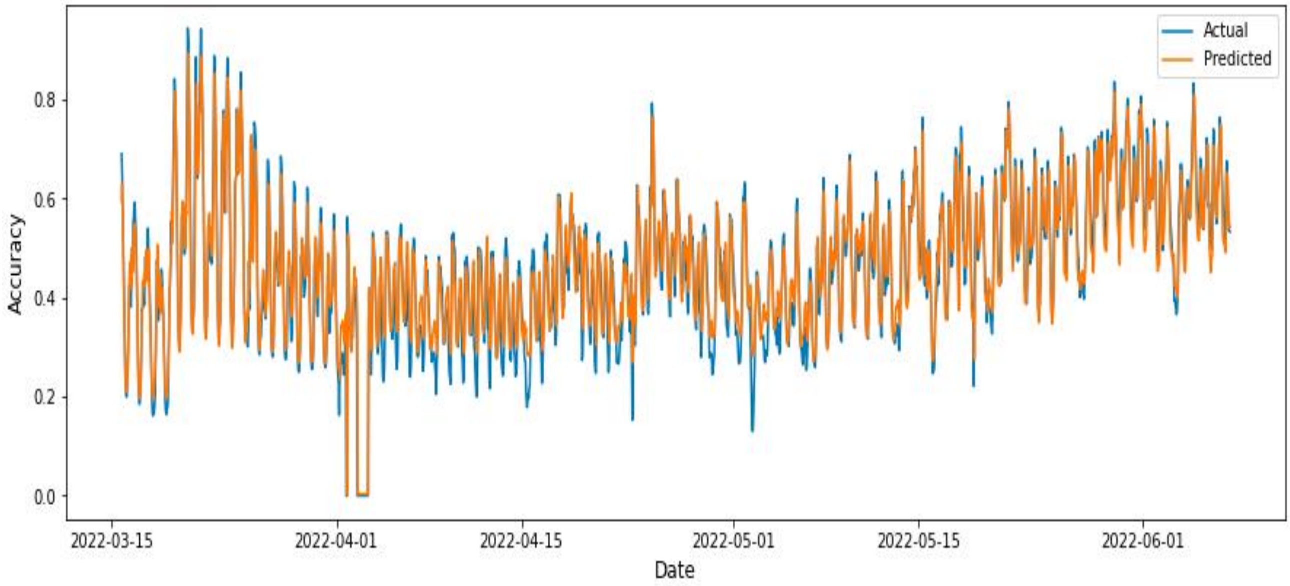

Figure 5.20: Gated recurrent unit result for forecasting electricity load value.

Figure 5.20 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.01, where R2 = 0.87749. It can be seen from Figure 5.20, the error rate in forecasting the loads on the peak is very small, but there is a very small difference in the minimum loads.

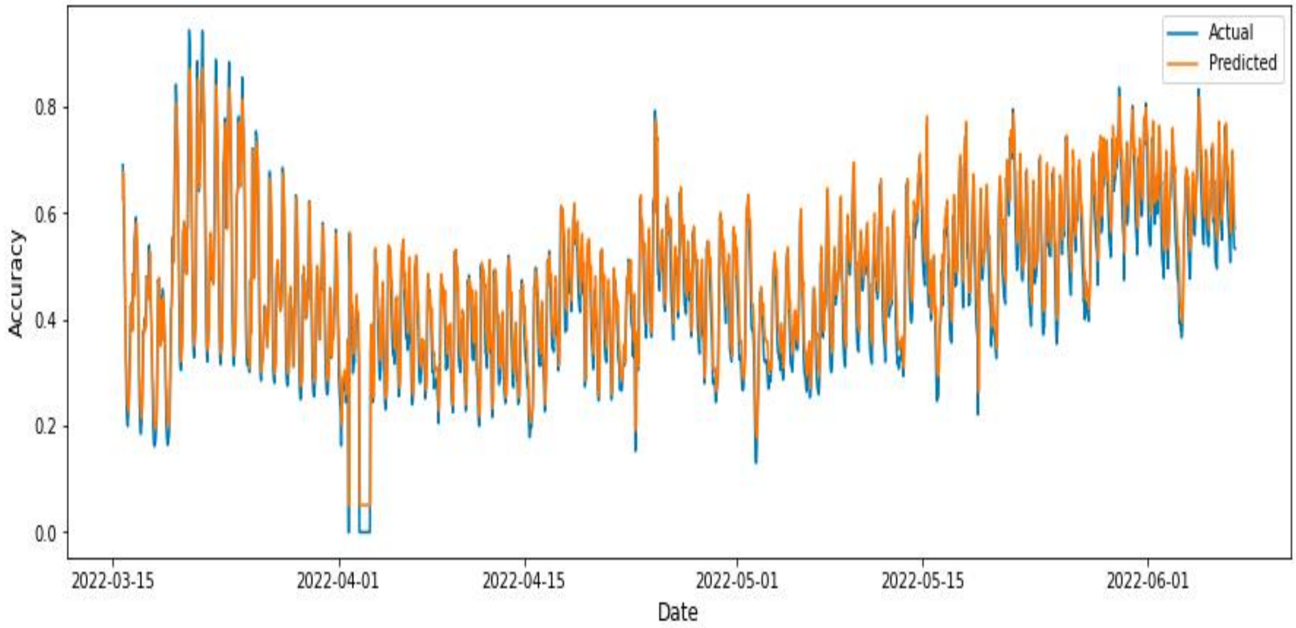

Figure 5.21: Recurrent neural network result for forecasting electricity load value.

Figure 5.21 shows the discrepancy between the actual and predictive results to forecast the STLF of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the RNN model, it was found that the RNN model achieved the highest value at learning rate of 0.01, where $R^2 = 0.85114$. It can be seen from Figure 5.21, the error rate in forecasting the loads on the peak is very small, and the results are very close to the GRU model.

5.2.3.2 Two Hidden Layers

In this section, two hidden layers were applied in addition to the input and output layer on each model with the Dropout to reduce overfitting and improve model performance. Table 5.8 shows the results for each model that was applied to two hidden layers and a different learning rate.

Table 5.8: Result RMSprop optimizer with two hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	**LSTM**	0.0471	0. 134	0.2171	0.1978
0.01		0.0080	0.6367	0.0895	0.0749
0.001		0.0039	0.8216	0.0627	0.0493
0.1	**GRU**	0.0129	0.4146	0.1137	0.0881
0.01		0.0026	0.8804	0.0513	0.0378

0.001		0.0032	0.8520	0.0571	0.0410
0.1	**RNN**	0.2190	-8.908	0.4679	0.4460
0.01		0.0046	0.7915	0.0678	0.0562
0.001		0.0046	0.7889	0.0683	0.0556

Table 5.8 shows the test results after applying each model based on the hyperparameter mentioned above. Different learning rate was used for each model. The results for the LSTM model were the worst results on rate of learning equal to 0.1, and the best results were on rate of learning equal to 0.001 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.1 and the lowest error rate on the learning rate of 0.01. Almost the same was true for the RNN model, where rate of learning equal to 0.1 was the worst error rate, and at a rate of 0.01, it was the lowest rate of error achieved by the model. All in all, the best model that got the lowest error rate is GRU with rate of learning equal to 0.01.

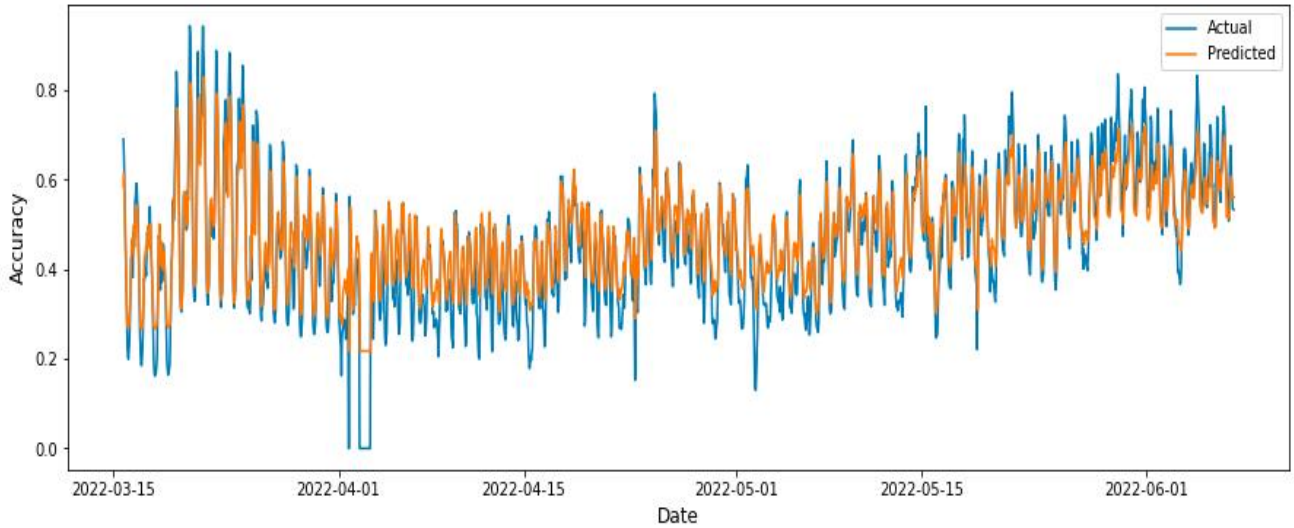

Figure 5.22: Long short-term memory result for forecasting electricity load value.

Figure 5.22 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.001, where R2 = 0.8216. It can be seen from Figure 5.22, the error rate in forecasting the loads on the peak is high, but there is a clear difference between the actual and forecasted values at the bottom and the peak loads (the electricity demand is minimal) in this model. Where the model was able to forecast the average loads in this case, as for the electrical loads at the top and the small loads, the model was not able to forecast their loads, and this led to a high error rate and a lack of accuracy for this model in this case.

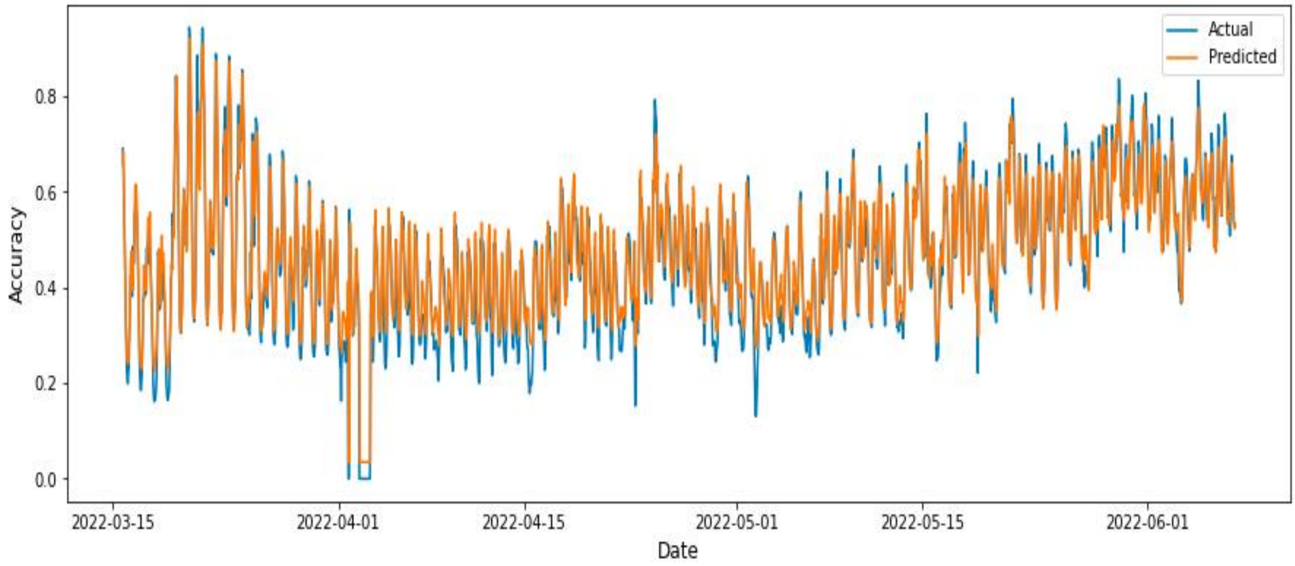

Figure 5.23: Gated recurrent unit result for forecasting electricity load value.

Figure 5.23 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.01, where $R^2 = 0.8804$. It can be seen from Figure 5.23, the error rate in forecasting the loads on the peak and minimum loads is very small.

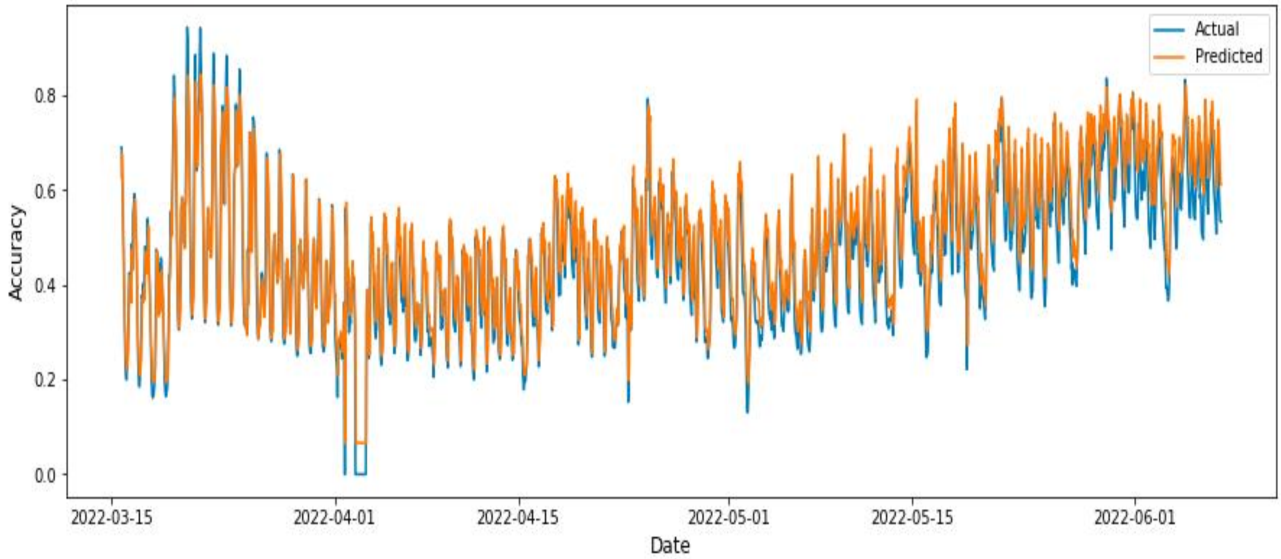

Figure 5.24: Recurrent neural network result for forecasting electricity load value.

Figure 5.24 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in the RNN model, it was found that the RNN model achieved the highest value at learning rate 0.01, where R2 = 0.7915. It can be seen from Figure 5.24, is a clear difference between the actual and forecasted values at the bottom (the electricity demand is minimal) are small in this model, and in the same case in the last test period, there is a fluctuation in the difference between the two values.

5.2.3.3 Three Hidden Layers

In this section, three hidden layers were applied in addition to the input and output layers on each model with the Dropout. Table 5.9 shows the results for each model applied to three hidden layers and a different learning rate.

Table 5.9: Result RMSprop optimizer with three hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	LSTM	0.03966	-0.7946	0.19916	0.16498
0.01		0.00422	0.80874	0.06501	0.04828
0.001		0.00683	0.69075	0.08267	0.06936
0.1		0.02307	-0.0438	0.15189	0.12041

0.01	**GRU**	0.00288	0.86941	0.05372	0.04146
0.001		0.00334	0.84857	0.05785	0.04172
0.1	**RNN**	0.02817	-0.2749	0.16786	0.13225
0.01		0.01341	0.39317	0.11581	0.09153
0.001		0.00961	0.56479	0.09807	0.08433

Table 5.9 shows the test results after applying each model based on the hyperparameter mentioned above. Different learning rate was used for each model. The results for the LSTM model were the worst results on rate of learning equal to 0.1, and the best results were on rate of learning equal to 0.01 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.1 and the lowest error rate on the learning rate of 0.01. Almost the same was true for the RNN model, where rate of learning equal to 0.1 was the worst error rate, and at a rate of 0.001, it was the lowest rate of error achieved by the model. All in all, the best model that got the lowest error rate is GRU with rate of learning equal to 0.01, and note that the results of the RNN model are very few of accuracy with a very high error rate.

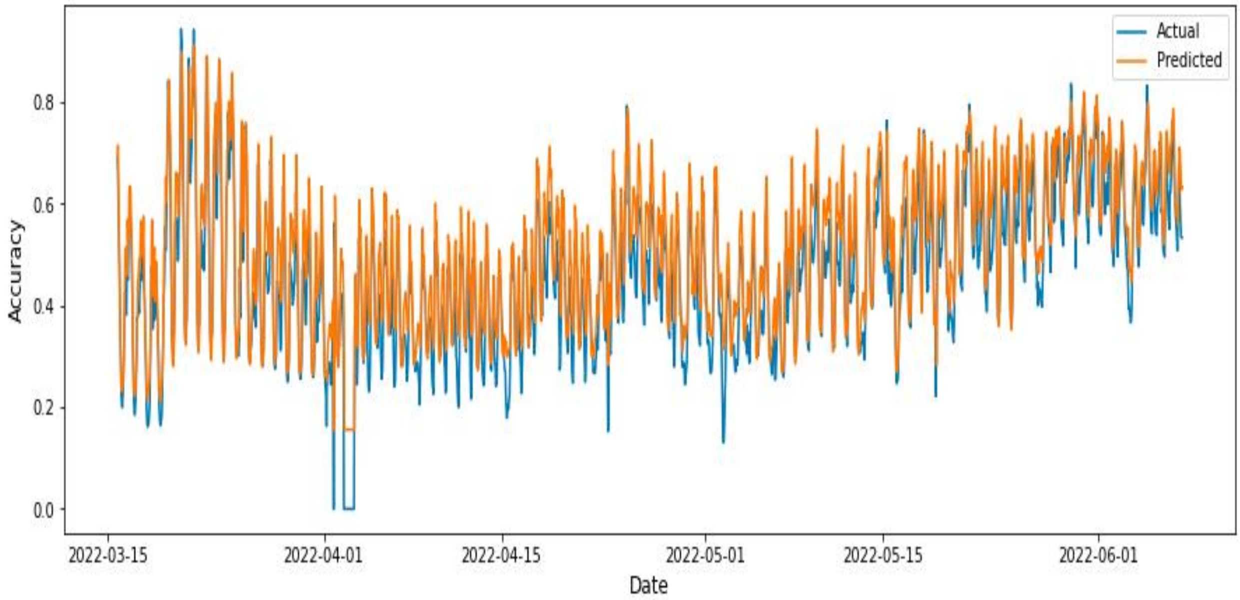

Figure 5.25: Long short-term memory result for forecasting electricity load value.

Figure 5.25 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.01, where R^2 = 0.80874. It can be seen from Figure 5.25, the error rate in forecasting the loads on the peak is high, but there is a clear difference between the actual and forecasted values at the bottom and the peak loads (the electricity demand is minimal) in this model. Where the model was able to forecast the average loads in this case, as for the electrical loads at the small loads, the model was not able to forecast their loads, and this led to a high error rate and a lack of accuracy for this model in this case.

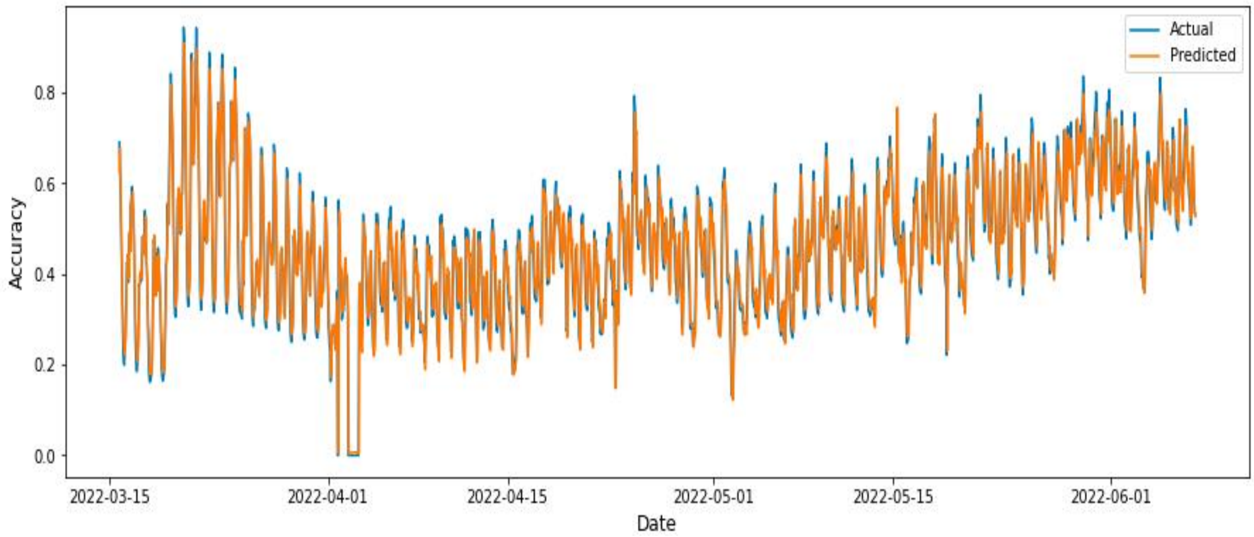

Figure 5.26: Gated recurrent unit result for forecasting electricity load value.

Figure 5.26 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.01, where R^2 = 0.86941. It can be seen from Figure 5.26, the error rate in forecasting the loads on the peak is small.

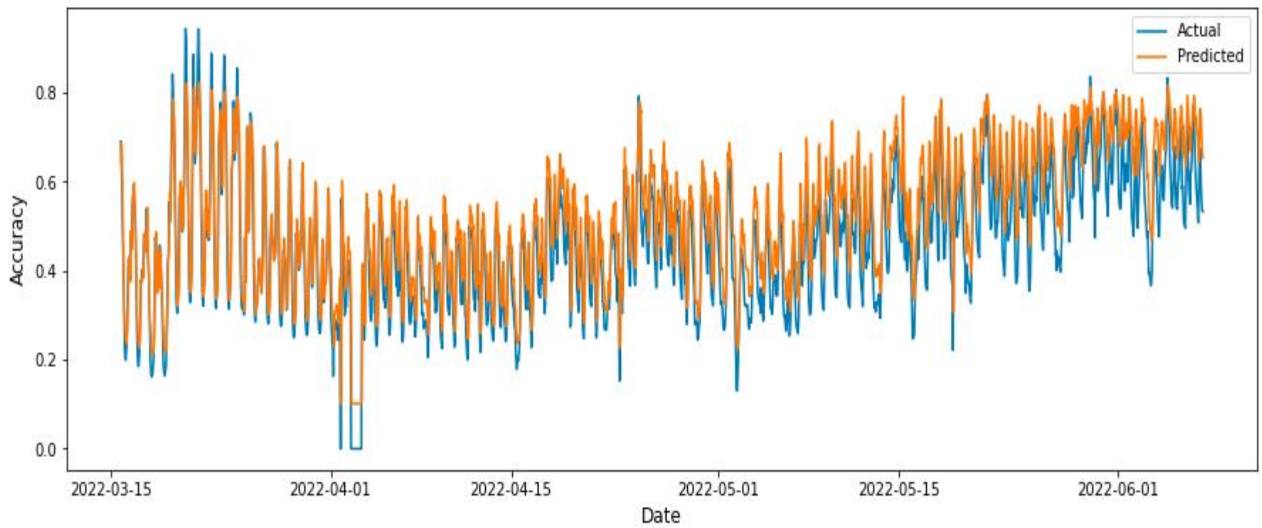

Figure 5.27: Recurrent neural network result for forecasting electricity load value.

Figure 5.27 shows the discrepancy between the actual and predictive results to forecast the STLF of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in RNN model, it was found that the RNN model achieved the highest value at learning rate 0.001, where R^2 = 0.85094. It can be seen from Figure 5.27, is a clear difference between the actual and forecasted values at the bottom (the electricity demand is minimal) in this model. In the same case in the peak value their a clear difference, And it can almost be said in this case that the model was unsuccessful in forecasting the electrical loads because there is a big difference between the actual and expected loads.

After applying the RMSprop optimizer in more than one way (one hidden layer, two hidden layers, three hidden layers, different learning rates) with machine learning models LSTM, RNN, and GRU. The results obtained from this optimizer are applied with two hidden layers With the GRU model and the learning rate is 0.01, it gave the best results, as the R^2 was 88.02%, RMSE is 0.0513, and the MAE was 0.0378.

5.2.4 Adadelta Optimizer

This section will show the results obtained after applying the AdaDelta optimizer. This optimizer monotonously reduces the decreasing learning rate based on a static moving window of gradient updates.

This optimizer was used with LSTM, RNN, and GRU machine learning models, to forecast electrical loads to get the lowest error rate of the model during the testing period. In this section, more than one hidden layer has been applied and tested, to compare them and choose the case with the lowest error and the highest accuracy.

5.2.4.1 One Hidden Layer

In this section, one hidden layer was applied in addition to the input and output layer on each model with the dropout in AdaDelta optimizer. Table 5.10 shows the results for each model that was applied to one hidden layer and a different learning rate.

Table 5.10: Result AdaDelta optimizer with one hidden layer.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	LSTM	0.00416	0.81143	0.06455	0.05274
0.01		0.01577	0.28612	0.12561	0.10147
0.001		0.16871	-6.6336	0.41075	0.38500
0.1	GRU	0.00292	0.86781	0.05405	0.04006
0.01		0.00959	0.56599	0.09794	0.07986
0.001		0.01934	0.12476	0.13908	0.10752
0.1	RNN	0.00301	0.86348	0.05492	0.04120
0.01		0.00586	0.73461	0.07658	0.06180
0.001		0.03744	-0.6942	0.19350	0.15291

Table 5.10 shows the test results after applying each model based on the hyperparameter mentioned above. A different learning rate was used for each model. The results for the LSTM model were the worst results on the rate of learning equal to 0.001, and the best results were on the rate of learning equal to 0.1 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.001 and the lowest error rate on the learning rate of 0.1. Almost the same was true for the RNN model, where the rate of learning equal to 0.001 was the worst error rate, and at a rate of 0.1, it was the lowest rate of error achieved by the model. All in all, the best model that got the lowest error rate is GRU with a rate of learning equal to 0.1 where R-square = 0.86781.

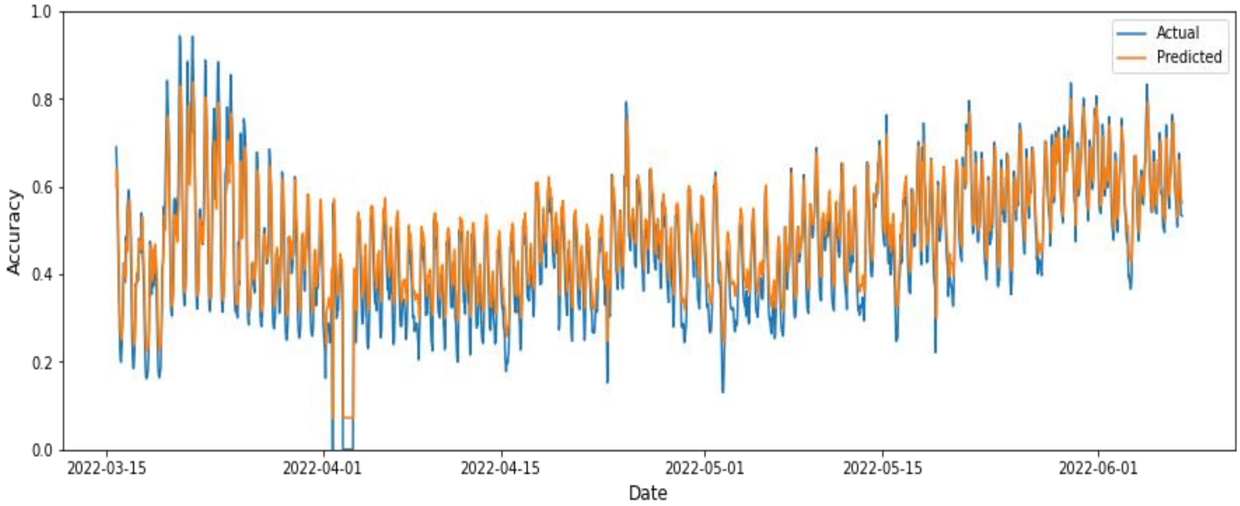

Figure 5.28: Long short-term memory result for forecasting electricity load value.

Figure 5.28 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.1, where R2 = 0.81143. It can be seen from Figure 5.28, the difference between the actual and expected values can be seen, especially in the peak values in which the loads are the highest, and there is a large and clear difference between the expected and actual values, which led to an increase in the error rate in this case when the loads are minimal.

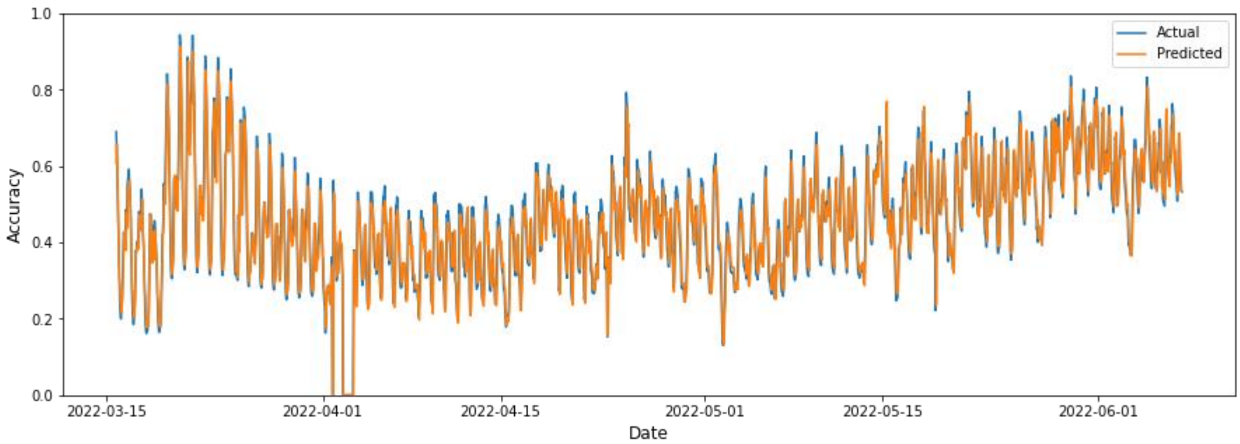

Figure 5.29: Gated recurrent unit result for forecasting electricity load value.

Figure 5.29 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate.

After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.1, where R^2 = 0.86781. It can be seen from Figure 5.29, the error rate in forecasting the loads on the peak is very small. In addition, in this case, this model got the best results compared to other models.

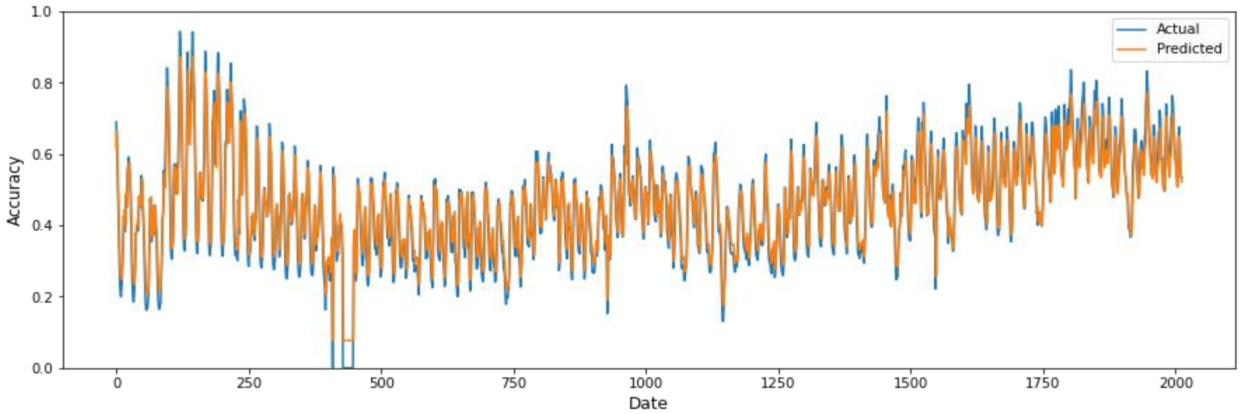

Figure 5.30: Recurrent neural network result for forecasting electricity load value.

Figure 5.30 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in RNN model, it was found that the RNN model achieved the highest value at learning rate 0.1, where R^2 = 0.86348. It can be seen from Figure 5.30, is a small difference between the actual and expected values in peak loads and low loads, and its results, in this case, are close to the results of the GRU model.

5.2.4.2 Two Hidden Layers

In this section, two hidden layers were applied in addition to the input and output layer on each model with the Dropout to reduce overfitting and improve model performance. Table 5.11 shows the results for each model that was applied to two hidden layers and a different learning rate.

Table 5.11: Result AdaDelta optimizer with two hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	**LSTM**	0.0060	0.7262	0.0777	0.0603
0.01		0.0188	0.1487	0.137-+1	0.1092
0.001		0.1843	-7.341	0.4293	0.4041
0.1	**GRU**	0.0029	0.8676	0.0540	0.0397
0.01		0.0129	0.4138	0.1138	0.0923
0.001		0.0293	-0.328	0.1713	0.1352
0.1	**RNN**	0.0034	0.8426	0.0589	0.0441
0.01		0.0132	0.3993	0.1152	0.0912
0.001		0.0264	-0.197	0.1626	0.1222

Table 5.11 shows the test results after applying each model based on the hyperparameter mentioned above. Different learning rate was used for each model. The results for the LSTM model were the worst results on rate of learning equal to 0.001, and the best results were on rate of learning equal to 0.1 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.001 and the lowest error rate on the learning rate of 0.1. Almost the same was true for the RNN model, where rate of learning equal to 0.001 was the worst error rate, and at a rate of 0.1, it was the lowest rate of error achieved by the model. As noted, the best model that got the lowest error rate is GRU with rate of learning equal to 0.1. All in all, there these results is that the lower the learning rate, the lower the accuracy and the greater the error rate.

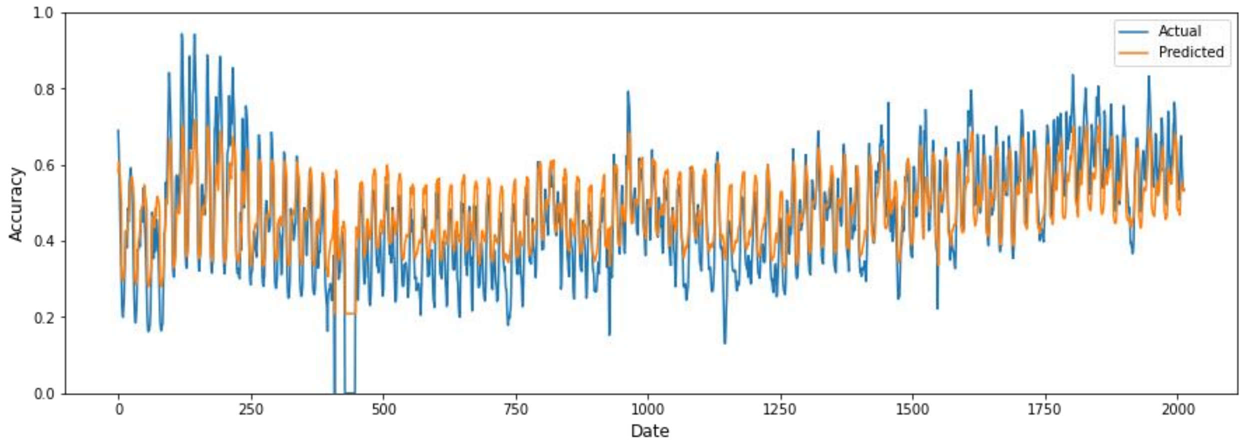

Figure 5.31: Long short-term memory result for forecasting electricity load value.

Figure 5.31 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.1, where R2 = 0.7262. It can be seen from Figure 5.31, the error rate in forecasting the loads on the peak is high, but there is a clear difference between the actual and forecasted values at the bottom and the peak loads (the electricity demand is minimal) in this model. Where the model was able to forecast the average loads in this case, as for the electrical loads at the top and the small loads, the model was not able to forecasting their loads, and this led to a high error rate and a lack of accuracy for this model in this case. The model in this case is considered a failure, and it cannot be relied upon in forecasting electrical loads because it has a large error rate and a low accuracy with a good estimate.

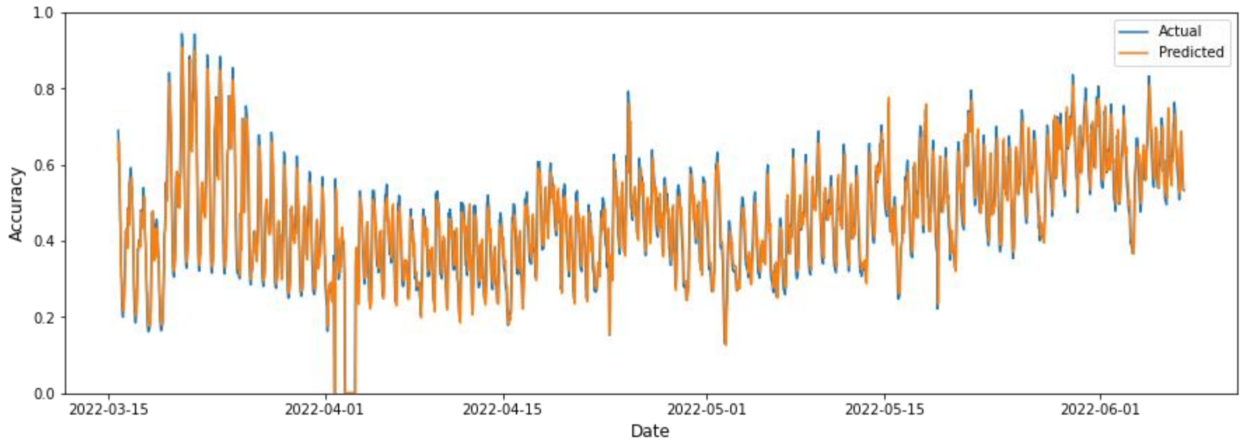

Figure 5.32: Gated recurrent unit result for forecasting electricity load value.

Figure 5.32 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.1, where $R^2 = 0.8676$. It can be seen from Figure 5.32, the error rate in forecasting the loads on the peak and minimum loads is very small.

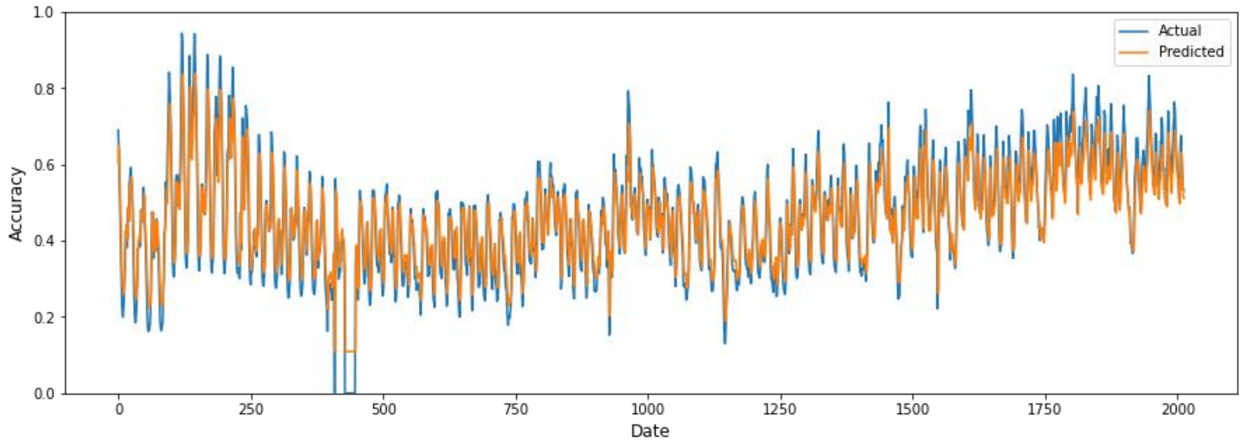

Figure 5.33: Recurrent neural network result for forecasting electricity load value.

Figure 5.33 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in RNN model, it was found that the

RNN model achieved the highest value at learning rate 0.1, where $R^2 = 0.8426$. It can be seen from Figure 5.33, is a slight difference between the actual and expected values in peak loads, especially at the end of the testing period, the difference was clear.

5.2.4.3 Three Hidden Layers

In this section, three hidden layers were applied in addition to the input and output layers on each model with the Dropout. Table 5.12 shows the results for each model applied to three hidden layers and a different learning rate.

Table 5.12: Result AdaDelta optimizer with three hidden layers.

Learning Rate	Model	MSE	R2	RMSE	MAE
0.1	**LSTM**	0.02018	0.03696	0.14205	0.11361
0.01		0.02253	-0.0198	0.15013	0.11925
0.001		0.19470	-7.8091	0.44124	0.41648
0.1	**GRU**	0.00292	0.86749	0.05411	0.04024
0.01		0.01215	0.45025	0.11022	0.09018
0.001		0.02445	-0.1063	0.15637	0.12286
0.1	**RNN**	0.00339	0.86348	0.05492	0.04120
0.01		0.01373	0.37867	0.11718	0.09260
0.001		0.05396	-1.4415	0.19737	0.19737

Table 5.12 shows the test results after applying each model based on the hyperparameter mentioned above. A different learning rate was used for each model. The results for the LSTM model were the worst results on the rate of learning equal to 0.001, and the best results were on the rate of learning equal to 0.1 where the error rate was the least possible. The results for the GRU model were the worst error rate on the learning rate of 0.001 and the lowest error rate on the learning rate of 0.1. Almost the same was true for the RNN model, where rate of learning equal to 0.001 was the worst error rate, and at a rate of 0.1, it was the lowest rate of error achieved by the model. As noted, the best model that got the lowest error rate is GRU with a rate of learning equal to 0.1. As it was noted that the shorter the learning period, the greater the error rate and the lower the accuracy, and this case is similar to the case of two hidden layers,

and the results are almost close to each other. All in all, the LSTM model was less accurate, or in other words, it did not give results, the forecasting results were almost non-existent and the error rate was higher.

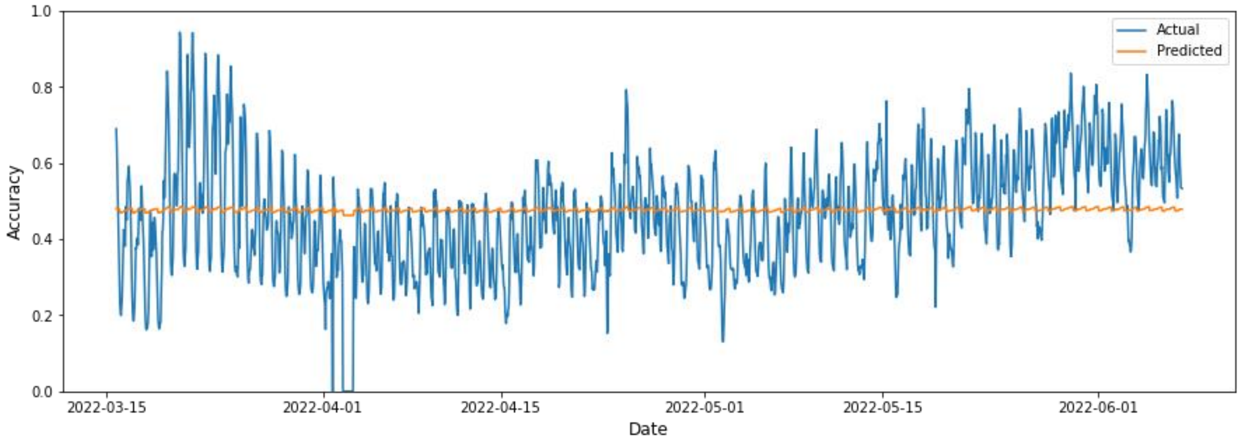

Figure 5.34: Long short-term memory result for forecasting electricity load value.

Figure 5.34 shows the discrepancy between the actual and predictive results to forecast the STLF of the LSTM model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in LSTM model, it was found that the LSTM model achieved the highest value at learning rate 0.1, where R^2 = 0.03696. It can be seen from Figure 5.34, the error rate in forecasting the loads is high, it was less accurate, or in other words, it did not give results, the forecast results were almost non-existent and the error rate was higher. In this case, the model could not predict, because the learning step was 0.1 and the model was unable to reach the global minimum of data to learn the model from all data, , therefore, the LSTM model does not converge.

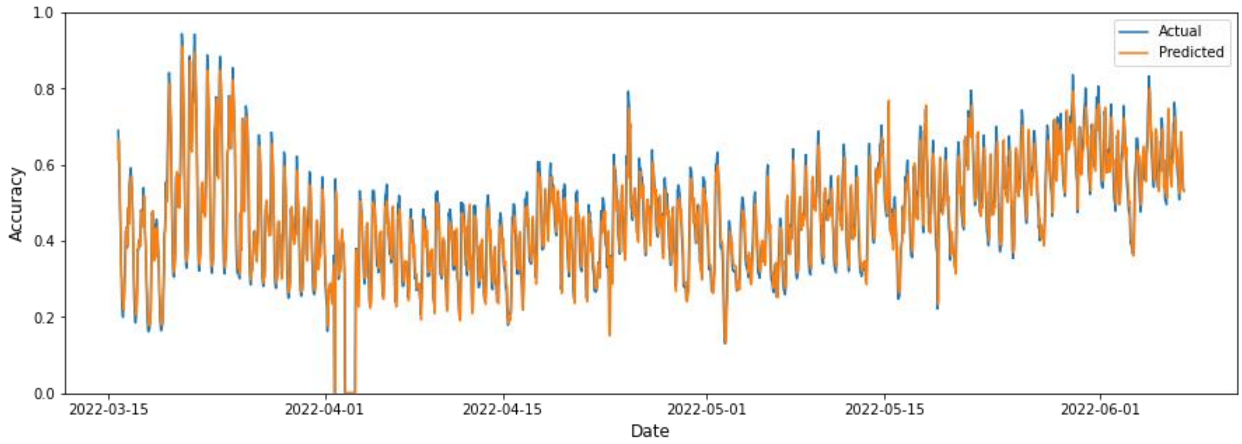

Figure 5.35: Gated recurrent unit result for forecasting electricity load value.

The result in figure 5.35 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the GRU model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in GRU model, it was found that the GRU model achieved the highest value at learning rate 0.1, where $R^2 = 0.86749$. It can be seen from Figure 5.35, the error rate in forecasting the loads on the peak is small.

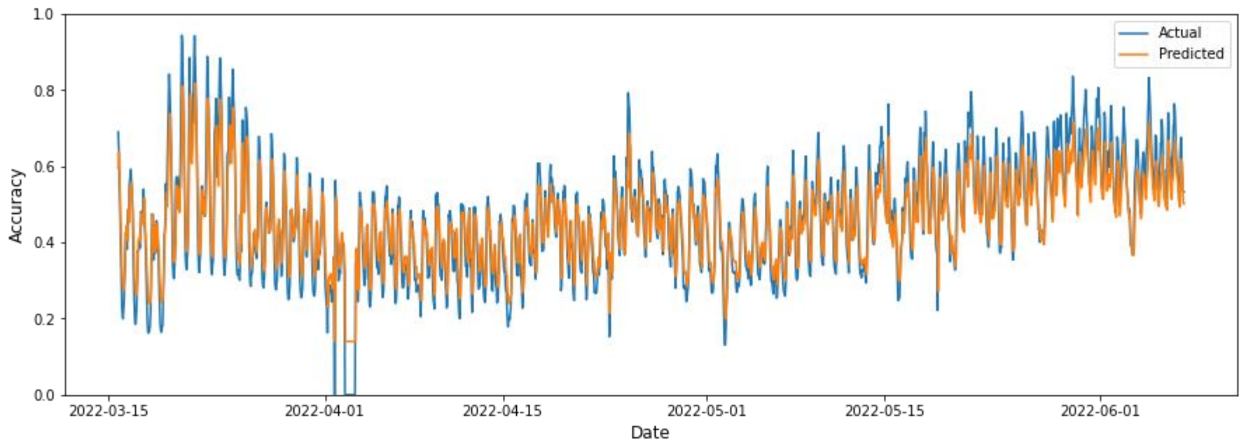

Figure 5.36: Recurrent neural network result for forecasting electricity load value.

The result in figure 5.36 shows the discrepancy between the actual and predictive results in order to forecast the STLF of the RNN model, where the test result values were taken from each learning rate. After comparing the test results for each learning rate in RNN model, it was found that the RNN model achieved the highest value at learning rate 0.1, where $R^2 = 0.86348$. It can be seen from Figure 5.36, is a clear difference between the actual and forecasting values at the bottom (the electricity demand is minimal) in this model. In the same case in the peak, value their clear difference. It can be seen there is a difference between the actual values and the expected values at the peak electrical loads, and the difference was very clear at the end of the test period, the model could not forecasting the electrical loads in the peak condition.

After applying the AdaDelta optimizer in more than one way (one hidden layer, two hidden layers, three hidden layers, different learning rates) with machine learning models LSTM, RNN, and GRU. The results obtained from this optimizer are applied with one hidden layer With the GRU model and the learning rate is 0.1. In other words, the results were close between the different classes with rate of learning equal to 0.1, it gave the best results, as the R^2 was 86.781%, RMSE is 0.05405, and the MAE was 0.04006.

5.3 Comparing DL Models Based On Mean Absolute Error

In this section, will show how MAE for each deep learning algorithm (LSTM, GRU, RNN). It will depend on a learning rate of 0.01 and two hidden layers, and every optimizer that was used because these parameters give the best results. Through MAE it will give an indication of which algorithm it will be relied upon to forecasting short-term electrical loads in addition to R squared, Figure 5.37 shows MAE for each model.

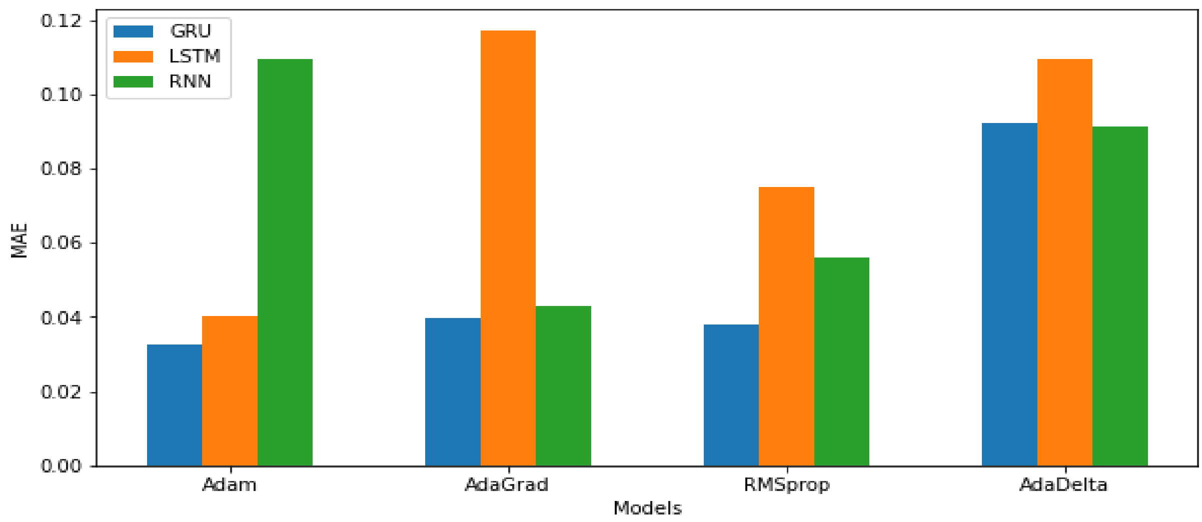

Figure 5.37: MAE results obtained by LSTM, RNN, and GRU models with more than one optimizer.

Figure 5.37 shows the results obtained by machine learning models from LSTM, RNN, and GRU on two hidden layers with a learning rate = 0.01. Here only taken the best results for comparison, and those applied to four optimizers (Adam, SGD, Adagrad, RMSprop, and AdaDelta), the best optimizer gave the lowest percentage of MAE, the Adam optimizer, and the AdaDelta optimizer was the worst in terms of the high error rate compared to the others. Figure 5.37 also shows that in Adam's optimizer, the GRU model was the best with an MAE of 0.03266.

5.4 Random Forest Forecasting Result.

This section presents the results of the random forest algorithm, in order to see whether deep learning models give better results than classical machine learning models or not. Table 5.13 presents the results of the model.

Table 5.13: Random Forest Result.

	MAE	RMSE	R^2
Random Forest (RF)	11.277	16.333	79.7889

Table 5.12 shows the test results after applying RF. The results for the RF model were RMSE= 16.333, MAE = 11.277, and R2= 84.

5.5 Conclusion

In this chapter, all the results obtained from the proposed deep-learning methods for forecasting short-term electrical loads are discussed and explained. Regression was based on R^2, MAE, RMSE, and MSE to compare machine-learning algorithms (LSTM, RNN, and GRU) and choose the best among them. In addition, applied more than one hidden layer, more than one learning rate, and four optimizations to choose the best model and get the lowest error rate. In addition, the RF algorithm was applied to compare with deep learning algorithms. Based on the results, the results of RF were lower than the results of deep learning.

The GRU model obtained the best results, where the R^2 value = 90.2% and MAE = 0.03266, RMSE = 0.04647. This is when applying Adam's optimizer and two hidden layers with a learning ratio of 0.01. Where many batch sizes and the number of epochs were tested, and the best batch size was 32 and the number of epochs was 50, with a training rate is 70% and a test is 30%, the best results were obtained. The results obtained from the models are promising and indicate that the electric load can be forecasted with high accuracy in Palestine. The following chapter provides conclusions and recommendations for future directions.

Chapter Six

Conclusion and Future Work

In this research, machine-learning algorithms, specifically deep learning on a large scale, were used to forecasting electrical loads based on the readings that were taken from the SCADA program for the connection point between the Israeli occupation and the Electricity Company of the Tubas area. The dataset used in this research is unique/first hand data as it is obtained from Tubas electric company for the first time to make data exploration and modeling. Load forecasting is the science of the economical amount of electric power for a utility company. It must be timely, as accurate as possible, reliable, and meaningful. This study also aims to bridge the gap between demand and energy, reduce electricity losses from large quantities, and plan the future of infrastructure more accurately.

Three deep learning algorithms (LSTM, GRU, and RNN) were used, in addition to real historical data obtained from Tubas Electricity Company. Where the GRU model proved to be the best in obtaining the highest accuracy and the lowest error rate. In addition, the RF algorithm was applied to forecasting the electrical load, where the result for this model did not satisfy. In addition, many improvements were used in building models, and the best one was Adam. As the model and data were applied to a hidden layer and more than one hidden layer, the use of two hidden layers in forecasting electrical loads was the best. The best GRU model achieved results as R^2 90.228%, MSE = 0.00215, MAE = .03266, and RMSE = 0.04647.

Simulation results show that temperature plays a large and influential factor in loads. Moreover, this research has proven that in the late night hours and early morning hours, consumption and loads are minimal, because all factories, institutions, and homes are closed. It is also shown that in the winter, the loads are low, because the irrigation of crops relies on rainwater, and there is no need to operate wells that depend on electricity. This research also has many practical implications. First, it created a model that can forecast the electric load for Palestine helps reduce power outages. Second, the positive effects of providing a more accurate STLF will reduce planning uncertainty. In addition, this research has proven that in the late night hours and early morning hours, consumption and loads are minimal, because all factories, institutions, and homes are closed.

In this research, the available features of Palestine electric company were taken and analyzed as discussed in the data description part. Based on our data exploration and explanation there are no sensors used in electric companies in Palestine to measure the humidity, since it is an essential metric for the electric load. This study recommended to install sensors in the electric companies to measure the humidity as it is an important measure for the electric load.

As a strategy for the future. It is advised that humidity data be collected and creating a model capable of forecasting electrical loads based on the available and recommended features, reading all consumption from homes through smart meters, and training the model online based on the data obtained from these smart meters. Which also helps us reveal the behavior of Palestinian families. In addition to building web interfaces that, display all the expected loads for the concerned department, while forecasting long-term electrical loads, because it needs data of no less than three years to forecasting these loads, and this data is not currently available for this period.

References

M. Amin and J. Stringer, “The electric power grid: Today and tomorrow,” MRS Bull., vol. 33, no. 4, pp. 399–407, 2008.

C.-T. Cheng, J.-J. Shen, X.-Y. Wu, and K. Chau, “Operation challenges for fast-growing China’s hydropower systems and respondence to energy saving and emission reduction,” Renew. Sustain. Energy Rev., vol. 16, no. 5, pp. 2386–2393, 2012.

H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural networks for short-term load forecasting: A review and evaluation,” IEEE Trans. power Syst., vol. 16, no. 1, pp. 44–55, 2001.

S. Soliman and A. Al-Kandari, “Electrical load forecasting: modeling and model construction,” Elsevier, vol. 5, no. 2, pp. 144–168, 2010.

E. A. Feinberg and D. Genethliou, “Load Forecasting,” Appl. Math. Restructured Electr. Power Syst., pp. 269–285, Jun. 2005.

T. Ranjan Jena, S. Sucharita Barik Assistant Professor, and S. Kumari Nayak Assistant Professor, “Electricity Consumption & Prediction using Machine Learning Models,” shabdbooks.com, vol. 9, no. 6, pp. 2804--2818, 2020.

S. V. Oprea, A. Bâra, F. C. Puican, and I. C. Radu, “Anomaly detection with machine learning algorithms and big data in electricity consumption,” Sustain., vol. 13, no. 19, 2021.

L. Lei, W. Chen, B. Wu, C. Chen, and W. Liu, “A building energy consumption prediction model based on rough set theory and deep learning algorithms,” Energy Build., vol. 240, p. 110886, Jun. 2021.

T. Liu, C. Xu, Y. Guo, and H. Chen, “A novel deep reinforcement learning based methodology for short-term HVAC system energy consumption prediction,” Int. J. Refrig., vol. 107, pp. 39–51, Nov. 2019.

H. Al-Bayaty, T. Mohammed, W. Wang, and A. Ghareeb, “City scale energy demand forecasting using machine learning based models: A comparative study,” ACM Int. Conf. Proceeding Ser., pp. 1--9, 2019.

T. Ahmad et al., “Supervised based machine learning models for short, medium and long-term energy prediction in distinct building environment,” Energy, vol. 158, pp. 17–32, Sep. 2018.

R. Geetha, K. Ramyadevi, and M. Balasubramanian, “Prediction of domestic power peak demand and consumption using supervised machine learning with smart meter dataset,” Multimed. Tools Appl. 2021 8013, vol. 80, no. 13, pp. 19675–19693, Mar. 2021.

C. Chen, Y. Liu, M. Kumar, J. Qin, and Y. Ren, “Energy consumption modelling using deep learning embedded semi-supervised learning,” Comput. Ind. Eng., vol. 135, pp. 757–765, Sep. 2019.

Z. A. Khan, M. Adil, N. Javaid, M. N. Saqib, M. Shafiq, and J. G. Choi, “Electricity Theft Detection Using Supervised Learning Techniques on Smart Meter Data,” Sustain. 2020, Vol. 12, Page 8023, vol. 12, no. 19, p. 8023, Sep. 2020, doi: 10.3390/SU12198023.

E. Prediction, C. Machine, and E. Learning, “Electricity Consumption & Prediction using Machine Learning Modelss,” Mukt Shabd J., vol. IX, no. 180303110001, pp. 2804–2818, 2020.

T.-Y. Kim and S.-B. Cho, “Predicting residential energy consumption using CNN-LSTM neural networks,” Energy, vol. 182, pp. 72–81, 2019.

Z. Wang and R. S.-R. and S. E. Reviews, “A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models,” Elsevier, vol. 14, pp. 33–60, 2017.

N. Mohan, K. P. Soman, and S. Sachin Kumar, “A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model,” Appl. Energy, vol. 232, pp. 229–244, Dec. 2018.

M. Lopez-Martin, A. Sanchez-Esguevillas, L. Hernandez-Callejo, J. I. Arribas, and B. Carro, “Novel Data-Driven Models Applied to Short-Term Electric Load Forecasting,” Appl. Sci. 2021, Vol. 11, Page 5708, vol. 11, no. 12, p. 5708, Jun. 2021.

M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, H. Abu-Rub, and F. S. Oueslati, “Short-term electric load forecasting based on data-driven deep learning techniques,” in IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, 2020, pp. 2565–2570.

N. Liu, Q. Tang, J. Zhang, W. Fan, and J. L.-A. Energy, “A hybrid forecasting model with parameter optimization for short-term load forecasting of micro-grids,” Elsevier, vol. 12, p. 25, 2014.

H. Willis and J. N.-G.-P. IEEE, “Spatial electric load forecasting: A tutorial review,” IEEE Access, vol. 71, pp. 232--253, 1983.

S. Ryu, J. Noh, and H. Kim, “Deep neural network based demand side short term load forecasting,” Energies, vol. 10, no. 1, pp. 1–20, 2017.

D. Ivanov, A. Tsipoulanidis, and J. Schönberger, Global Supply Chain and Operations Management. Cham: Springer International Publishing, 2017.

C. Kuster, Y. Rezgui, and M. Mourshed, “Electrical load forecasting models: A critical systematic review,” Sustain. Cities Soc., vol. 35, pp. 257–270, 2017.

S. Arora and J. W. Taylor, “Rule-based autoregressive moving average models for forecasting load on special days: A case study for France,” Eur. J. Oper. Res., vol. 266, no. 1, pp. 259–268, 2018.

H. Takeda, Y. Tamura, and S. S. Energy, “Using the ensemble Kalman filter for electricity load forecasting and analysis,” Elsevier, vol. 8, pp. 95–125, 2018.

S. Maldonado, A. Gonzalez, and S. C.-A. S. Computing, “Automatic time series analysis for electric load forecasting via support vector regression,” Elsevier, pp. 235–268, 2019.

J. Rendon-Sanchez, L. de M.-E. J. of Operational, and U. 2019, “Structural combination of seasonal exponential smoothing forecasts applied to load forecasting,” Eur. J. Oper. Res., vol. 275, pp. 916–924, 2019.

K. Lindberg, P. Seljom, H. Madsen, D. F.-U. Policy, and U. 2019, “Long-term electricity load forecasting: Current and future trends,” Elsevier, vol. 58, pp. 102–119, 2019.

T. Hong and S. F. Forecasting, “Probabilistic electric load forecasting: A tutorial review,” Int. J. Forecast., vol. 32, pp. 914–938, 2016.

S. Kloker, T. Straub, and C. Weinhardt, “Designing a crowd forecasting tool to combine prediction markets and real-time delphi,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10243 LNCS, pp. 468–473, 2017.

B. Goehry and Y. Goude, “Aggregation of multi-scale experts for bottom-up load forecasting,” ieeexplore.ieee.org, vol. 11, pp. 1895–1904, 2019.

S. H. Rafi, N. Al-Masood, S. R. Deeba, and E. Hossain, “A short-term load forecasting method using integrated CNN and LSTM network,” IEEE Access, vol. 9, pp. 32436–32448, 2021.

L. Alhmoud and Q. Nawafleh, “Short-term load forecasting for Jordan’s Power System Using Neural Network based Different Optimization Techniques,” Proc. - 2019 IEEE Int. Electr. Eng., pp. 1–29, Jun. 2019.

Z. Shafiei Chafi and H. Afrakhte, “Short-Term Load Forecasting Using Neural Network and Particle Swarm Optimization (PSO) Algorithm,” Math. Probl. Eng., vol. 2021, 2021.

H. Eskandari, M. Imani, and M. P. Moghaddam, “Convolutional and recurrent neural network based model for short-term load forecasting,” Electr. Power Syst. Res., vol. 195, p. 107173, Jun. 2021.

L. Yin and J. Xie, “Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems,” Appl. Energy, vol. 283, p. 116328, Feb. 2021.

N. Andriopoulos et al., “Short term electric load forecasting based on data transformation and statistical machine learning,” mdpi.com, vol. 13, pp. 540–596, 2020.

L. Alhmoud, R. Abu Khurma, A. M. Al-Zoubi, and I. Aljarah, “A Real-Time Electrical Load Forecasting in Jordan Using an Enhanced Evolutionary Feedforward Neural Network,” Sensors 2021, Vol. 21, Page 6240, vol. 21, no. 18, p. 6240, Sep. 2021.

S. Acharya, Y. Wi, J. L.- Energies, and U. 2019, “Short-term load forecasting for a single household based on convolution neural networks using data augmentation,” mdpi-Energies, vol. 12, p. 3560, 2019.

X. Gao et al., “Short-term electricity load forecasting model based on EMD-GRU with feature selection,” mdpi.com, vol. 4, p. 29, 2020.

B. Yuan, B. He, J. Yan, J. Jiang, Z. Wei, and X. Shen, “Short-term electricity consumption forecasting method based on empirical mode decomposition of long-short term memory network,” in IOP Conference Series: Earth and Environmental Science, 2022, vol. 983, no. 1, p. 12004.

S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, “Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches †,” Energies 2018, Vol. 11, Page 1636, vol. 11, no. 7, p. 1636, Jun. 2018.

F. He, J. Zhou, Z. kai Feng, G. Liu, and Y. Yang, “A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm,” Appl. Energy, vol. 237, pp. 103–116, Mar. 2019.

X. B. Jin et al., “Deep-Learning Forecasting Method for Electric Power Load via Attention-Based Encoder-Decoder with Bayesian Optimization,” Energies 2021, Vol. 14, Page 1596, vol. 14, no. 6, p. 1596, Mar. 2021.

L. Gong, M. Yu, S. Jiang, V. Cutsuridis, and S. Pearson, “Deep Learning Based Prediction on Greenhouse Crop Yield Combined TCN and RNN,” Sensors 2021, Vol. 21, Page 4537, vol. 21, no. 13, p. 4537, Jul. 2021.

L. Han et al., “Enhanced deep networks for short-term and medium-term load forecasting,” Ieee Access, vol. 7, pp. 4045--4055, 2018.

Z. Wang, Y. Wang, R. Zeng, … R. S.-E. and, and U. 2018, “Random Forest based hourly building energy prediction,” Elsevier-Energy Build., vol. 171, pp. 11--25, 2018.

M. Ahmad, M. Mourshed, Y. R.-E. and Buildings, and U. 2017, “Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption,” Elsevier-Energy Build., vol. 147, pp. 77--89, 2017.

A. Lahouar and J. Ben Hadj Slama, “Day-ahead load forecast using random forest and expert input selection,” Energy Convers. Manag., vol. 103, pp. 1040–1051, Oct. 2015.

N. M. M. Bendaoud and N. Farah, “Using deep learning for short-term load forecasting,” Neural Comput. Appl., vol. 32, no. 18, pp. 15029–15041, Sep. 2020.

M. Imani, “Electrical load-temperature CNN for residential load forecasting,” Energy, vol. 227, p. 120480, Jul. 2021.

K. Grolinger, M. A. M. Capretz, and L. Seewald, “Energy consumption prediction with big data: Balancing prediction accuracy and computational resources,” in 2016 IEEE International Congress on Big Data (BigData Congress), 2016, pp. 157–164.

M. Bilgili, N. Arslan, A. Sekertekin, and A. Yasar, “Application of long short-term memory (LSTM) neural network based on deep learning for electricity energy consumption forecasting,” Turkish J. Electr. Eng. Comput. Sci., vol. 30, no. 1, pp. 140–157, 2022.

J. Wang, X. Chen, F. Zhang, F. Chen, and Y. Xin, “Building Load Forecasting Using Deep Neural Network with Efficient Feature Fusion,” J. Mod. Power Syst. Clean Energy, vol. 9, no. 1, pp. 160–169, Jan. 2021.

M. Alobaidi, F. Chebana, and M. M. Energy, “Robust ensemble learning framework for day-ahead forecasting of household based energy consumption,” Elsevier, pp. 215–248, 2018.

N. Somu, G. R. M R, and K. Ramamritham, “A hybrid model for building energy consumption forecasting using long short term memory networks,” Appl. Energy, vol. 261, p. 114131, Mar. 2020.

M. Fekri, H. Patel, K. Grolinger, and V. S. Energy, “Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network,” Elsevier, vol. 25, 2021.

L. Li, K. Ota, and M. D.- Systems, “Everything is image: CNN-based short-term electrical load forecasting for smart grid,” ieeexplore.ieee.org, pp. 344--351, 2017.

H. Shi, M. Xu, and R. L.- Grid, “Deep learning for household load forecasting—A novel pooling deep RNN,” IEEE Trans. Smart Grid, pp. 133–190, 2017.

D. M. Amarasinghe, K, “Deep neural networks for energy load forecasting,” ieeexplore.ieee.org, pp. 1483--1488, 2017.

K. Bache and M. Lichman, “UCI machine learning repository,” IEEE Access, p. 23, 2013.

W. Kong, Z. Dong, Y. Jia, D. Hill, and Y Xu, “Short-term residential load forecasting based on LSTM recurrent neural network,” IEEE Trans. Smart Grid, pp. 841--851, 2017.

M. A. R. Biswas, M. D. Robinson, and N. Fumo, “Prediction of residential building energy consumption: A neural network approach,” Energy, vol. 117, pp. 84–92, Dec. 2016.

N. Thokala and A Bapna, “A deployable electrical load forecasting solution for commercial buildings,” IEEE-Industrial Technol. (ICIT), pp. 1101--1106, 2018.

M. Bessani, J. A. D. Massignan, T. M. O. Santos, J. B. A. London, and C. D. Maciel, “Multiple households very short-term load forecasting using bayesian networks,” Electr. Power Syst. Res., vol. 189, p. 106733, Dec. 2020.

F. Alasali, K. Nusair, L. Alhmoud, and E. Z. Sustainability, “Impact of the covid-19 pandemic on electricity demand and load forecasting,” Sustainability, vol. 13, p. 1435, 2021.

T. Ahmad and H. Chen, “Potential of three variant machine-learning models for forecasting district level medium-term and long-term energy demand in smart grid environment,” Energy, vol. 160, pp. 1008–1020, Oct. 2018.

E. A. Feilat, “Talal Al-Sha’abi D., Mo-mani MA Long-term load forecasting using neural network approach for Jordan’s power system,” Eng. Press, vol. 1, pp. 43–50, 2017.

N. Shirzadi, A. Nizami, M. Khazen, and M. Nik-Bakht, “Medium-Term Regional Electricity Load Forecasting through Machine Learning and Deep Learning,” Des. 2021, Vol. 5, Page 27, vol. 5, no. 2, p. 27, Apr. 2021.

A. Rahman, V. Srikumar, and A. S. Energy, “Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks,” Appl. Energy, vol. 212, pp. 372--385, 2018.

R. K. Agrawal, F. Muchahary, M. Tripathi, R. Kumar Agrawal, and M. M. Tripathi, “Long term load forecasting with hourly predictions based on long-short-term-memory networks,” ieeexplore.ieee.org, pp. 1--6, 2018.

S. R. Khuntia, J. L. Rueda, and M. A. M. M. van Der Meijden, “Forecasting the load of electrical power systems in mid-and long-term horizons: a review,” IET Gener. Transm. \& Distrib., vol. 10, no. 16, pp. 3971–3977, 2016.

U. Javed et al., “Exploratory Data Analysis Based Short-Term Electrical Load Forecasting: A Comprehensive Analysis,” Energies, vol. 14, no. 17, p. 5510, 2021.

A. G. Asuero, A. Sayago, and A. G. González, “The correlation coefficient: An overview,” Crit. Rev. Anal. Chem., vol. 36, no. 1, pp. 41–59, Jan. 2006.

C. C. Aggarwal and others, Data mining: the textbook, vol. 1. Springer, 2015.

A. Jain, K. Nandakumar, and A. Ross, “Score normalization in multimodal biometric systems,” Pattern Recognit., vol. 38, no. 12, pp. 2270–2285, 2005.

H. Vafaie and K. A. De Jong, “Genetic Algorithms as a Tool for Feature Selection in Machine Learning.,” in ICTAI, 1992, pp. 200–203.

A. Norouzi, M. Aliramezani, and C. R. Koch, “A correlation-based model order reduction approach for a diesel engine NOx and brake mean effective pressure dynamic model using machine learning,” Int. J. Engine Res., vol. 22, no. 8, pp. 2654–2672, 2021.

F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” Neural Comput., vol. 12, no. 10, pp. 2451–2471, 2000.

I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

C. Fan, J. Wang, W. Gang, and S. Li, “Assessment of deep recurrent neural network-based strategies for short-term building energy predictions,” Appl. Energy, vol. 236, pp. 700–710, 2019.

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv Prepr. arXiv1412.3555, 2014.

O. Dürr, B. Sick, and E. Murina, Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability. Manning Publications, 2020.

K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” arXiv Prepr. arXiv1409.1259, vol. 12, pp. 125–196, 2014.

D. Britz, “Recurrent neural network tutorial, part 4 implementing a gru/lstm rnn with python and theano,” Inf. Syst. E-bus. Manag., pp. 560–587, 2015.

M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Light gated recurrent units for speech recognition,” IEEE Trans. Emerg. Top. Comput. Intell., vol. 2, no. 2, pp. 92–102, 2018.

Y. Su and C.-C. J. Kuo, “On extended long short-term memory and dependent bidirectional recurrent neural network,” Neurocomputing, vol. 356, pp. 151–161, 2019.

N. Gruber and A. Jockisch, “Are GRU cells more specific and LSTM cells more sensitive in motive classification of text,” Front. Artif. Intell., vol. 3, p. 40, 2020.

B. Veloso, J. Gama, B. Malheiro, and J. Vinagre, “Hyperparameter self-tuning for data streams,” Inf. Fusion, vol. 76, pp. 75–86, 2021.

J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization.," J. Mach. Learn. Res., vol. 12, no. 7, 2011.

D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv Prepr. arXiv, pp. 1–42, 2014.

G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," Cited on, vol. 14, no. 8, p. 2, 2012.

M. D. Zeiler, "Adadelta: an adaptive learning rate method," arXiv Prepr. arXiv, pp. 1–56, 2012.

K. P. Murphy, Machine learning: a probabilistic perspective. 2012.

B. Delyon, "Stochastic approximation with decreasing gain: Convergence and asymptotic theory," Unpubl. Lect. notes, Univ. Rennes, vol. 26, pp. 1–36, 2000.

R. Chopade, A. Stanam, A. Narayanan, and S. Pawar, "Cyclical Learning Rates (CLR's) for Improving Training Accuracies and Lowering Computational Cost," IEEE Access, vol. 15, pp. 198–235, 2021.

J. Patterson and A. Gibson, Deep learning: A practitioner's approach. 2017.

S. Ruder, "An overview of gradient descent optimization algorithms," arXiv Prepr. arXiv, vol. 18, pp. 1–96, 2016.

Y. Nesterov, Introductory lectures on convex optimization: A basic course, vol. 87. Springer Science \& Business Media, 2003.

H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, “Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting,” Water, vol. 12, no. 5, p. 1500, 2020.

M. C. Tjepkema-Cloostermans, R. C. V de Carvalho, and M. J. A. M. van Putten, “Deep learning for detection of focal epileptiform discharges from scalp EEG recordings,” Clin. Neurophysiol., vol. 129, no. 10, pp. 2191–2196, 2018.

S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V Le, “Don’t decay the learning rate, increase the batch size,” arXiv Prepr. arXiv, vol. 19, pp. 582–618, 2017.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv Prepr. arXiv, no. 6, pp. 1–35, 2012.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

D. Warde-Farley, I. J. Goodfellow, A. Courville, and Y. Bengio, “An empirical analysis of dropout in piecewise linear networks,” arXiv Prepr. arXiv, vol. 15, no. 4, pp. 890–945, 2013.

V. Plevris, G. Solorzano, N. P. Bakas, and M. E. A. Ben Seghier, “investigation of performance metrics in regression analysis and machine learning-based prediction models,” ieee trans. emerg. top. comput. intell., vol. 13, no. 6, pp. 1–40, 2021.

الملخص

يعد التنبؤ بمتطلبات الحمل الكهربائي المستقبلية الخطوة الأولى في تصميم نظام الطاقة ونموه. يُعد توقع الحمل أمرًا بالغ الأهمية من الناحيتين الفنية والمالية. من الضروري في الإجراءات التجارية التقليدية وتخطيط نقل الكهرباء تحسين أداء نظام الطاقة والموثوقية والسلامة والاستقرار مع خفض تكاليف التشغيل أيضًا. نظرًا لأهميتها في إدارة الطاقة وتخطيط البنية التحتية والميزانية ، فقد أثار التنبؤ بحمل الكهرباء اهتمام الباحثين والشركات.

تم الحصول على بيانات الحمل الكهربائي الفعلي من شركة توزيع الكهرباء في فلسطين - شركة كهرباء محافظة طوباس لمدة عام تقريبًا. تستند هذه البيانات إلى الأحمال الكهربائية التي حصلت عليها شركة الكهرباء من نقطة التوصيل مع الجانب الإسرائيلي ، لإجمالي الطلب على الأحمال في كل دقيقة. الهدف الرئيسي من هذا العمل هو إجراء حساب سهل ودقيق للتنبؤ بالحمل الكهربائي بناءً على قياسات الأحمال الحالية في فلسطين. لأن عدم اليقين في التنبؤ بالأحمال الكهربائية قد يؤدي إلى هدر الأموال والموارد ، خاصة في فلسطين لعدم وجود بحث أو دراسة للتنبؤ بهذه الأحمال.

نماذج الانحدار - تم استخدام التعلم العميق للتنبؤ بالأحمال الكهربائية باستخدام LSTM و GRU و RNN.حيث تم الاعتماد عليها للحصول على أفضل النتائج وأقل معدل خطأ مقارنة بالدراسات السابقة ، تم تعديل معلمات الضبط بناءً على عدة طرق ، على النحو التالي: معدل التعلم ، نوع المحسنات ، نسبة التدريب والاختبار ، وظائف التنشيط ، الدُفعة الحجم والعصر ، عدد الطبقات المخفية. بعد تجربة وتطبيق جميع طرق ضبط المعلمات ، حصل نموذج GRU على أفضل نموذج من حيث الدقة وأقل معدل خطأ ، عند تطبيق طبقتين مخفيتين ومحسِّن آدم ، بمعدل اختبار 70٪ وعدد من epochs = 50 وحجم الدُفعة 32.

أظهرت النتائج أن نموذج GRU حقق R squared بنسبة 90.228٪ ، و MSE = 0.00215 ، و MAE = 0.03266.