T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A NEW FORECASTING SYSTEM FOR ELECTRICAL LOADS IN THE MIDDLE EUPHRATES REGION USING NEURAL NETWORKS

MSc THESIS

Mudher ABDULHADI

Computer and Information Engineering Department

Computer Engineering Program

MARCH 2025

T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A NEW FORECASTING SYSTEM FOR ELECTRICAL LOADS IN THE MIDDLE EUPHRATES REGION USING NEURAL NETWORKS

MSc THESIS

Mudher ABDULHADI

Computer and Information Engineering Department

Computer Engineering Program

Thesis Advisor: Prof. Dr. Ahmet ZENGİN

MARCH 2025

The thesis work titled “A New Forecasting System for Electrical Loads in The Middle Euphrates Region Using Neural Networks” prepared by Mudher Abdulhadi was accepted by the following jury on 21/03/2025 by unanimously/majority of votes as a MSc Thesis in Sakarya University Graduate School of Natural and Applied Sciences, Computer and Information Engineering department, Computer and Information Engineering program.

Thesis Jury

Head of Jury :	**Prof. Dr. Ahmet ZENGİN** Sakarya University	....
Jury Member :	**Prof. Dr. Devrim AKGÜN** Sakarya University	....
Jury Member :	**Dr. A.F.M. Suaib AKHTER** Sakarya Applied Science University	....

STATEMENT OF COMPLIANCE WITH THE ETHICAL PRINCIPLES AND RULES

I declare that the thesis work titled "A NEW FORECASTING SYSTEM FOR ELECTRICAL LOADS IN THE MIDDLE EUPHRATES REGION USING NEURAL NETWORKS", which I have prepared in accordance with Sakarya University Graduate School of Natural and Applied Sciences regulations and Higher Education Institutions Scientific Research and Publication Ethics Directive, belongs to me, is an original work, I have acted in accordance with the regulations and directives mentioned above at all stages of my study, I did not get the innovations and results contained in the thesis from anywhere else, I duly cited the references for the works I used in my thesis, I did not submit this thesis to another scientific committee for academic purposes and to obtain a title, in accordance with the articles 9/2 and 22/2 of the Sakarya University Graduate Education and Training Regulation published in the Official Gazette dated 20.04.2016, a report was received in accordance with the criteria determined by the graduate school using the plagiarism software program to which Sakarya University is a subscriber, I accept all kinds of legal responsibility that may arise in case of a situation contrary to this statement.

(21/03/2025)

Mudher ABDULHADI

To my parents, brother, sisters, wife and children…

ACKNOWLEDGMENTS

I would like to express my deep gratitude and appreciation to my advisor, Prof. Dr. Ahmet Zengin, for his big heart, his continuous support that spared no effort to help me in this thesis, and his patience and valuable guidance.

I would also like to express my sincere appreciation to my professors at the university, and college and the teaching staff who have imparted their knowledge and experience to me. I thank all my friends for their continuous support throughout the difficulties.

I would also like to express my deep gratitude to my family and extended family who have inspired me.

Thank you from the bottom of my heart.

Mudher ABDULHADI

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANN	: Artificial Neural Network
CNN	: Convolutional Neural Network
DNN	: Deep Neural Network
IoT	: Internet of Things
KNN	: K-Nearest Neighbors
KWh	: Kilowatt-hour
LSTM	: Long Short-Term Memory
MAE	: Mean Absolute Error
MAPE	: Mean Absolute Percentage Error
ML	: Machine Learning
MLP	: Multilayer Perceptron
MSE	: Mean Squared Error
MW	: Megawatt
RMSE	: Root Mean Squared Error
RNN	: Recurrent Neural Network
SVM	: Support Vector Machine

LIST OF TABLES

Page

LIST OF FIGURES

Page

A NEW FORECASTING SYSTEM FOR ELECTRICAL LOADS IN THE MIDDLE EUPHRATES REGION USING NEURAL NETWORKS

SUMMARY

This study develops neural network models for electrical load forecasting in Babylon Governorate, Iraq, focusing on Feedforward Neural Networks (FNN) and Long Short-Term Memory (LSTM) networks.

Load forecasting is essential for managing power supplies and systems because it can guide the optimal allocation of resources, planning the right time for maintenance and reducing energy costs among others. Load forecasting is of great importance due to the complexity of power systems and the variability in demand. Statistical methods such as ARIMA have been used in traditional forecasting techniques. Although they work well in some circumstances, they do not incorporate all non-linear structures between variables. To address this, neural networks, especially those based on deep learning, have proven useful tools by enhancing forecasting accuracy.

This thesis aims to use “FNN and LSTM”, a form of artificial neural networks to forecast electrical load demand. The data analyzed in this research includes load data for 33/11 kV power distribution stations in Babil Governorate, Iraq, from January 2021 to April 2024. Temperature information was also collected from reliable sources such as Accuweather, Wunderground, and Weather. Load data and weather conditions were integrated into the model. The main activity is to forecast the electrical load for 2024 and 2025 with special attention to the effects of retaining fluctuations in demand during distinct months. Some of the data used in this research include monthly electrical load demand records as well as temperature records during the same period. The load data consists of energy consumption of several feeding stations located throughout the governorate, and temperature data is also used to understand the difference in energy use with the help of weather conditions. The first process was data cleaning and relevant data were formatted to improve data quality.

MinMaxScaler and standardization methods were used where the data were normalized to scale for input into the neural network model. Normalization was applied, specifically, to make the ranges of all features equal because it is mandatory for the convergence of neurons during training.

Model Design and Training, two different artificial neural network models were applied: the feedforward neural network (FNN) and the long short-term memory (LSTM) network. The aim is to analyze data and predict outcomes based on the nature of the available datasets. The selection of the optimal model depends on several factors, including data structure, model accuracy, and training efficiency.

The FNN and LSTM models were compared, and loss and validation plots were constructed by tracing their respective loss curves during training and validation. The model was constructed, and model summaries were created. The FNN consists of dense layers with dropout regularization to prevent overfitting, resulting in a relatively

lightweight architecture (approximately 11,521 parameters). However, the LSTM includes specialized layers for sequential data (such as LSTM and embedding), making it more complex (approximately 79,375 parameters). This complexity enhances its suitability for temporal or textual data but requires greater computational resources than the FNN.

To validate and avoid overtraining the model, the Early Stopping technique was used to monitor the performance and stop training when the performance improvement stopped. The data obtained was divided into training and test datasets, where 80% of the data was trained and 20% was tested. Special methods, including the Dropout layer, were introduced to enhance the generality so that the model does not depend too much on the patterns in the training database. Moreover, hyperparameter tuning was also performed to fine-tune the "model performance," including learning rate, batch size, and number of epochs. The prediction accuracy of the neural network model constructed in the present work was measured by Mean Square Error in the test data (MSE) Mean Absolute Error (MAE) and Mean Percent Absolute Error (MAPE). These metrics provide a good indicator to evaluate the model's ability to predict unseen data and identify areas that deserve further improvement.

The first set of results indicated that the model captured the general behavior of the electrical load demand pattern but with varying degrees of accuracy in different months. For example, there were higher demands during the summer, especially in July and August, as people used more air conditioning in various residential facilities. Similarly, demand in the winter months of January and February was higher due to heating systems. However, as temperatures dropped in April, May, October and November, the modeled demand was observed to adjust or decline. The model was particularly sensitive to these seasonal differences, hence the need to incorporate correct weather data into subsequent versions.

Our results show a large approximation to the true values, differences in some months due to climate change, and large differences between summer and winter temperatures; temperatures range from a maximum of around 50 degrees in summer to a minimum of 5 degrees in winter. This is a large difference. This resulted in the test data evaluation being an approximate MSE of 95,000, MAPE of 18% in the test data evaluation. This indicates that the model can highly accurately predict and is useful in capturing broader trends. There are certain months that it is expected to produce lower forecast errors, and more factors should be considered when developing the model. More complex neural network architectures should be included in future studies to obtain better results due to some advantages of processing long series. Including more variables across multiple sectors, such as economic indicators, energy prices and consumption patterns, would make the demand model more accurate. The techniques used to produce ensemble estimates are generally superior to individual models because they can reduce variance and bias and thus produce more reliable estimates.

Although the model has been shown to perform well, work and development in this area always need to be developed to keep pace with the increase in electrical loads and human growth; the model can be improved by incorporating neural network structures such as recurrent neural network / long short-term memory (RNN/LSTM) that can handle time series such as electrical load demand. These structures could cope with long-term dependencies in the data, which is necessary for forecasting over a long period. Another recommendation is to try adding more variables, including standard of living, economic factors, energy costs and energy consumption rates by different residential, industrial and commercial sectors. The first set of results indicated that the model captured the general behavior of the electrical load demand pattern but with varying degrees of accuracy in different months. For example, there were higher demands during the summer, especially in July and August, as people used more air conditioning in various residential facilities. Similarly, demand in the winter months of January and February was higher due to heating systems. However, as temperatures dropped in April, May, October and November, the modeled demand was observed to adjust or decline. The model was particularly sensitive to these seasonal differences, hence the need to incorporate correct weather data into subsequent versions.

ORTA FIRAT BÖLGESİNDEKİ ELEKTRİK YÜKLERİ İÇİN YAPAY SİNİR AĞLARI KULLANARAK YENİ BİR TAHMİN SİSTEMİ TASARIMI

ÖZET

Bu çalışma, ileri beslemeli sinir ağları (FNN'ler) ve uzun-kısa süreli bellek (LSTM) ağlarına odaklanarak Babylon bölgesinde elektrik yükü tahmini için yapay sinir ağı modelleri geliştirmeyi amaçlamaktadır.

Yük tahmini, optimum kaynak tahsisi, doğru bakım planlaması ve enerji maliyetinin azaltılmasında yol gösterici bir rol oynayabileceğinden enerji kaynaklarının ve elektrik sistemlerinin yönetiminde çok önemlidir. Enerji sistemlerinin karmaşıklığı ve sürekli değişen talep göz önüne alındığında, yük tahmini büyük önem taşımaktadır.

ARIMA gibi bazı istatistiksel yöntemler bazı durumlarda iyi performans gösterse de, geleneksel tahmin teknikleri değişkenler arasındaki doğrusal olmayanlıkları tam olarak yakalamaz. Öte yandan, özellikle derin öğrenmeye dayalı olan sinir ağları, tahmin doğruluğunu iyileştirmek için yararlı araçlar olduklarını kanıtlamıştır.

Bu tez, elektrik yükü talebini tahmin etmek için bir tür yapay sinir ağı olan FNN ve LSTM'yi kullanmayı amaçlamaktadır. Bu araştırmada analiz edilen veriler, Ocak 2021'den Nisan 2024'e kadar olan dönemde Irak'ın Babil bölgesindeki 33/11 kV elektrik dağıtım istasyonlarından gelen yük verilerini içerir.

Sıcaklık bilgileri ayrıca Accuweather, Wonderground ve Weather gibi güvenilir kaynaklardan toplandı. Yük ve hava durumu verileri, 2024 ve 2025 için elektrik yükünü tahmin etmek üzere belirli aylardaki talep dalgalanmalarının etkisine özel olarak odaklanılarak modele dahil edildi.

Bu araştırmada kullanılan verilerin bir kısmı, sıcaklık kayıtlarına ek olarak aynı döneme ait aylık yük talebi kayıtlarını içerir. Yük verileri, bölgeye dağılmış çeşitli elektrik santrallerindeki enerji tüketiminden oluşur ve sıcaklık verileri, hava koşullarının enerji tüketimi üzerindeki etkisini anlamak için kullanılır.

Veri temizliğinden sonra, veri kalitesi optimizasyon ve biçimlendirme yoluyla iyileştirildi. Birkaç aya ait veriler, maksimum ve minimum sıcaklıkları girdi olarak ve elektrik yükü talebini hedef olarak belirterek pandas kullanılarak tek bir veri çerçevesinde birleştirildi. Ayrıca, aykırı değerleri önlemek için hedef değerler budandı.

MinMaxScaler gibi normalizasyon yöntemleri ve standardizasyon yöntemleri, giriş verilerini sinir ağı modeline ölçeklemek için kullanılır. Normalizasyon, özellikle nöronların eğitim sırasında birleşmesini sağlamak için önemlidir, bu nedenle özellik aralıklarını tekdüze hale getirmek için uygulanır.

Modeli tasarlamak ve eğitmek için iki farklı yapay sinir ağı türü kullanıldı: ileri beslemeli sinir ağları (FNN'ler) ve uzun-kısa süreli bellek (LSTM'ler). Amaç, verileri mevcut doğasına göre analiz etmek ve sonuçları tahmin etmektir. En uygun modeli

seçmek, veri yapısı, model doğruluğu ve eğitim verimliliği gibi çeşitli faktörlere bağlıdır.

Yapay sinir ağı modelini oluşturmak için TensorFlow ve Keras kullanıldı. Model doğruluğunu sağlamak ve aşırı öğrenmeyi önlemek için, performansı izlemek ve optimizasyon durduğunda eğitimi durdurmak için erken durdurma kullanıldı. Birkaç eğitim döneminden sonra öğrenme oranını kademeli olarak azaltmak için bir öğrenme oranı zamanlayıcısı kullanıldı ve bu da eğitimi sabitlemeye yardımcı oldu. Adam optimize edicisi de kullanıldı.

Veriler bir eğitim seti (%80) ve bir test seti (%20) olarak ayrıldı. Genelleştirmeyi artırmak ve eğitim verisi desenlerine olan bağımlılığı azaltmak için, bırakma katmanları gibi teknikler kullanıldı. Performans ayrıca öğrenme oranı, parti boyutu ve eğitim dönemi sayısı gibi hiperparametreler seçilerek de ayarlandı.

Bu çalışmada kullanılan modelin tahmin doğruluğu, ortalama kare hatası (MSE), ortalama mutlak hata (MAE) ve ortalama mutlak yüzde hatası (MAPE) gibi metrikler kullanılarak test verileri üzerinde değerlendirildi. Bu metrikler, modelin görülmeyen verileri tahmin etme ve iyileştirme alanlarını belirleme yeteneğinin iyi bir göstergesidir.

Ön sonuçlar, modelin elektrik yükü talebinin genel davranışını ay bazında değişen doğrulukla doğru bir şekilde yakalayabildiğini gösterdi. Örneğin, konut tesislerinde klima ünitelerinin kullanımının artması nedeniyle yaz aylarında, özellikle Temmuz ve Ağustos aylarında talepte bir artış gözlemlendi. Benzer şekilde, ısıtma sistemleri nedeniyle Ocak ve Şubat aylarında talep arttı. Ancak, daha düşük sıcaklıklarla Nisan, Mayıs, Ekim ve Kasım aylarında talep azaldı veya sabitlendi. Model, bu mevsimsel değişikliklere karşı yeterince hassastı ve modelin gelecekteki sürümlerine doğru hava durumu verilerinin dahil edilmesi ihtiyacını vurguladı.

Sonuçlar, gerçek değerlerle yüksek derecede yakınsama gösterdi ve bazı aylarda iklim değişikliği nedeniyle değişiklikler ve yaz ve kış sıcaklıkları arasında önemli değişiklikler görüldü. Sıcaklıklar yazın yaklaşık 50°C'den kışın minimum 5°C'ye kadar değişiyordu. Test verisi değerlendirmesi yaklaşık 95.000'lik bir MSE ve %18'lik bir MAPE üretti.

Uygulamadan sonra, FNN ve LSTM modelleri karşılaştırıldı, eğitim ve doğrulama sırasında kayıp eğrileri izlendi ve kayıp ve doğrulama grafikleri oluşturuldu. FNN modeli yoğun katmanlardan oluşur ve nispeten basit bir yapı (yaklaşık 11.521 parametre) ile sonuçlanır. Öte yandan LSTM modeli, sıralı verilere (LSTM ve yerleştirme gibi) ayrılmış katmanlar içerir ve bu da onu daha karmaşık hale getirir (yaklaşık 79.375 parametre). Bu karmaşıklık, zamansal veya metinsel veriler için uygunluğunu artırır, ancak FNN'ye kıyasla daha yüksek bir hesaplama maliyeti gerektirir.

Bu sonuçlar, modelin oldukça doğru tahminler yapma yeteneğine sahip olduğunu ve hatanın daha düşük olması beklenen belirli aylarla genel eğilimleri izlemek için çok yararlı olduğunu göstermektedir. Ayrıca, model geliştirilirken ek faktörlerin dikkate alınması gerektiğini ve gelecekteki çalışmalarda daha karmaşık sinir ağı yapılarının kullanılmasının uzun zaman serilerini ele alma yetenekleri nedeniyle daha iyi sonuçlar verebileceğini vurgulamaktadır.

Ekonomik göstergeler, enerji fiyatları ve tüketim kalıpları gibi farklı sektörlerden daha fazla değişkenin dahil edilmesi, talep modelini daha doğru hale getirecektir. Toplu tahminler üretmek için kullanılan toplama teknikleri, varyansı ve önyargıyı azaltarak daha güvenilir tahminler elde edilmesiyle sonuçlandığı için bireysel modellerden de üstündür.

1. INTRODUCTION

Electrical energy is the most important factor, and its availability and continuity are vital for modern industrial and economic development. Electrical load forecasting plays an important role in stabilizing and optimizing efficiency in electricity networks since accurate load forecasting allows electricity suppliers to manage and maintain generation plants, distribute electricity optimally, and avoid power failures. This means that accurate forecasts also reduce operating expenses and increase efficiency in utilizing the available natural resources by embracing renewable energy technologies.

Since many and varied factors and interactions constantly change, such as consumption patterns, new technologies or the climate, more accurate and time-effective models are required for these forecasts of electrical loads due to their capability of handling large volumes of mixed data and adapting to complex patterns.

Artificial intelligence, specifically neural networks and machine learning, have appeared to be capable of providing sound solutions to the complexity and variability of load forecasting [1,2]. These networks have shown great ability in making new pattern recognition on large datasets, especially for load forecasting [3,4].

ANNs are triggered by the concept of brain neurons and are computing models that can learn through data [2]. Lately, several variants of neural networks, including CNNs, RNNs, and LSTMs, have been introduced with their specific solutions for certain issues with data and prediction[5], [6], [7].

CNN/LSTM has attained high publicity from users due to its function of feature learning from raw pattern data; this function is important in image identification and time series processes [5], [7]. For datasets with a time component, such as electrical load forecasting, RNNs and LSTMs are often used because they are designed to capture temporal dependencies[6], [8].

The approach has been modified to weigh the different neural networks and produce a hybrid model to enhance the efficiency prediction and resilience [9], [10]. Also, the advance in more superior neural networks, such as deep learning, where several layers of processing are employed, has boosted the efficiency of such models to work with more complex and non-linear types of data [7], [11].

This research aims to review the advancement in the use of neural networks for electrical load forecasting in the last decade and compare and contrast the various neural network models.

Moreover, the research will employ and advance electrical load forecasting models that use machine learning and deep neural networks to enhance the prediction results and reduce errors. In this study, historical electrical load data and data concerning factors affecting the load will allow for constructing a valuable prediction model, which, in turn, will be evaluated for appropriate performance criteria. As a result, this thesis will enhance actual applications of electrical load estimation in actual systems and provide novel, valuable information to power corporations and the industry.

1.1. Background and Motivation

Electric load forecasting is vital in controlling and scheduling electric power systems to meet supply and demand. Load forecasting is important for several tasks as it allows for stabilizing the power grid, planning operational activities, and integrating renewable power sources [12], [13]. Previously, time series analysis, autoregressive models, and regression analysis have been used to forecast load. However, these approaches may not effectively capture the intrinsic relationship patterns of load data because of its nature of time and nonlinear as the power grid becomes more complex over time and time-varying due to various factors such as weather conditions, economic activities, and consumer behavior [14], [15].

Load forecasting has changed dramatically in recent years with the improvement of data mining techniques, such as machine learning and neural networks, where complex load patterns have been explored with more accuracy and robustness. Today's neural networks, especially deep learning models, have performed impressively well in dealing with big data, identifying complex patterns, and predicting significantly [16], [17]. They are extremely useful for load forecasting because they can establish non-linear relations. The CNN, RNN, and LSTM networks are among the architecture that has already proved fruitful in this domain [18], [19].

The general purpose of this research is derived from the urgency of developing more robust and effective load forecasting models that could capture the dynamism of power systems. The mobility of renewable sources of electricity has led to fluctuations in the generation and consumption of electricity in the systems. This is a major problem for classical power forecasting methods, which require constructing complex models for load forecasting under various conditions [17], [20]. To resolve these problems, this study will employ neural networks and help enhance the current load forecasting models in the future.

Furthermore, integrating smart technologies into the grid and abundant access to big data opened a new avenue to improve the models' accuracy. Load forecasting can be enhanced through the use of smart meters, sensors in IoT devices and other related gadgets that produce ample data[21], [22]. Neural networks with competency in processing data and learning from large data collections are well-suited to benefit from such data sources so that better decisions are made regarding the operation of power systems [21], [22].

1.2. Problem Statement

In the utility context, load forecasting is important because it allows the power system to properly balance load with supply by anticipating demand. However, conventional techniques for load forecasting have several drawbacks where autoregressive models are mainly used, which cannot capture the complex and nonlinear patterns associated with load data [23], [24]. As renewable resources are gaining importance in electricity production, and new challenges are arising from the complexity of power systems and their developments, the use of advanced methods for better prediction is called for.

ANNs have been identified as suitable substitutions to other approaches since they yield more accurate results regarding the non-linear interactions embedded in load data [25], [26]. Different types of ANNs have been applied in load forecasting [10], [27]. However, deep learning models such as CNNs and RNNs outperform others since they capture temporal dependencies and even sophisticated patterns inherent in the data. Nevertheless, several issues arise when adopting these models in real-life load forecasting systems; these include the requirement for vast amounts of labeled data and the complexity associated with the high computational costs and interpretability of the models at work [28], [29]. In response to these challenges, the current research will create and deploy adaptive neural network models optimized for electrical load prediction. This paper will therefore examine the effectiveness of deep learning approaches such as integrating CNNs with LSTM networks that would enhance the accuracy and reliability of the load forecast. In addition, the study will also explore the applicability of transfer learning and other methods to lower the amount of data and processing power needed to train deep learning models for this task [30], [31].

1.3. Objectives of the Study

As stated earlier, this work's main objective is to build a complex, accurate, dependable model based on neural networks for electric load forecasting. Given the performance of feed-forward networks (FNNs) and LSTM networks in handling time series data, this study proposes to enhance load forecasting by employing the strengths of feed-forward neural networks (FNNs), recurrent neural networks (RNNs), and LSTM networks. The following are the precise goals:

Design and implementation: Feedforward networks (FNNs), recurrent neural networks (RNNs), and LSTM network models are used to build a new load forecasting model by leveraging their respective strengths in modeling load data's temporal and geographical characteristics.

- Performance Evaluation: Compare the output of the suggested model with some of the state-of-the-art machine learning models and the traditional forecasting models. This involves considering the degree to which the model is scalable, accurate, and efficient regarding computer usage.
- Data Integration: To incorporate big data from smart meters, sensors, and other IoT devices to enhance the forecast accuracy of the model. This objective centers on how several data inputs can be combined to improve recognition accuracy under different load scenarios.
- Applicability to Real-World Scenarios: The following were postulated as key objectives of the work:

To enhance the general understanding of the interconnectedness between grid stability management, operational planning and integration of renewable energy systems using the proposed model.

To showcase the applicability of the developed model in real-life cases in grid stability management, operational planning, and integration of renewable energy systems. This involves checking the regional dispatch with different grid conditions and even with different percentages of renewable energy integration.

- Contribution to the Field: To make a valuable and significant input to both the theoretical body of knowledge and the engineering practice concerning load forecasting through artificial neural networks and to shed light on the possibilities and challenges of applying such models in today's power systems.

1.4. Organization of the Dissertation

The structure of this dissertation is aimed at covering all necessary aspects related to the load forecasting using neural networks, and the main emphasis is given to the description of the architecture of different neural networks' architectures and methods when applied to the electrical load forecasting.

Chapter 1 has the background and justification for the research, the problem statement, and the objectives. A brief and general account of how the present dissertation is structured is also given.

Chapter 2 examines the literature on load forecasting techniques with special emphasis on the use of neural networks. It describes Feed-forward neural networks, Recurrent neural networks, and Convolutional neural networks and their use in load forecasting. This chapter also discusses prior studies on feature selection, data pre-processing, and assessment in the context of load forecasting.

Chapter 3 Data preprocessing: The clean-up process of the load data is discussed. Some of the data preprocessing techniques discussed include removing the errors in the collected load data, transforming the data and normalizing the data. The chapter also provides information on the selection of the model parameters, training processes and the evaluation criteria used in the model.

Chapter 4 gives a detailed explanation of the data utilized in the study, the type of measurements taken, and general information about the data collected. It also contains the so-called Exploratory Data Analysis (EDA) to find the significant relationships and patterns for load forecasting. This chapter also outlines the neural networks utilized in load forecasting, focusing on their development and application. It explains the architecture of the model, the steps adopted in training the model, and the methods employed to enhance the model's performance. The chapter also explains the components used in the development of the system using software tools and libraries.

Chapter 5 aims to provide the results derived from the load forecasting models and discuss them. The models are then assessed based on several measures, and the results are triangulated with those of conventional load forecasting techniques. Based on the results of the analysis, the chapter also addresses the impact of the findings and their significance to the area of power systems.

Chapter 6 briefly recaps the main conclusions made during this research work, assesses the contribution of the current thesis to the existing field of knowledge, and explores possible future study directions. Finally, the chapter's considerations concern the research's methodological restrictions and the possibilities for their enhancement in further work.

2. LITERATURE REVIEW

2.1. Overview of Load Forecasting Methods

Load forecasting is an essential function used in the management of electrical power systems, and it is useful in determining the expected load in the power system. In the years that have passed, several forecasting techniques have been invented, all of which have advantages and weaknesses concerning the categorization of methods. It is found that they fall under three categories, namely, classical statistics methods, artificial intelligence techniques, and a combination of both.

2.1.1. Traditional statistical methods

The other conventional load forecasting techniques include Time Series Analysis, Regression Models, and Exponential Smoothing. Time series approaches such as ARIMA (Auto Regressive Integrated Moving Average) and SARIMA (Seasonal ARIMA) have been popular since they can capture linear relationships in the past load data [32]. Regression models predict load values for the future, given some independent variables such as temperature, time, and economic variables. Exponential smoothing, such as Holt-Winters, is worth mentioning among all the smoothing techniques, as it effectively identifies trends and cycles in the data samples [15].

2.1.2. Machine learning approaches

More recently, due to an increase in computational capability and availability of data, load forecasting has embraced the application of machine learning techniques. SVM, ANN or DTs are important methodologies used in electricity demand forecasting [13], [20]. Among them, Deep Learning models, especially CNNs and RNNs, have higher accuracy in identifying the complex non-linear relationships between the data variables and are hence suitable for short-term load forecasting [21], [22].

2.1.3. Hybrid models

To increase the reliability of the load forecasts, models that consist of both statistical and machine-learning approaches have been suggested. These models utilize the advantages inherent to this or that approach to enhance the prediction quality.

2.1.4. Proficient methods and trends

Some current studies have centered their research on incorporating Big Data Analytics & IoT data into the load forecasting equations. These approaches seek to enhance the accuracy of the forecasts by integrating actual smart meters and other sensor data into the models [17]. Furthermore, more attention has been paid to Probabilistic Load Forecasting (PLF), which delivers not only a point but a set of possible future demand values along with the corresponding probabilities of occurrence [13].

2.2. Application of Neural Network for Load Forecasting

Neural networks have been most useful in load forecasting because they can capture nonlinearities that may not be apparent from simple load evaluations using conventional statistical methods. The various forms of the networks have been applied in many forms with certain characteristics that are more suitable for certain aspects of load forecasting.

2.3. Feedforward Neural Networks (FNNs)

Fully Connected Feedforward Neural Networks (FNNs) are the neural networks mostly used in artificial neural networks where there is more of a linear interconnection of the nodes with no loop connections. They have been particularly applied in load forecasting because of their simple structure and capability of reflecting past load information and future load estimates. FNNs operate in the following manner: the input data is fed forward to the input layer, then through one or more hidden layers, and out to the output layer, where they are fully interconnected. They are especially suitable in short-size load forecasts when the system's relatively stable state [24], [33]. However, the performance of FNNs has been proven satisfactory for load forecasting studies by applying techniques such as feature selection and data normalization [26], [34]. The general concept of feedforward neural networks is shown in (Figure 2.1).

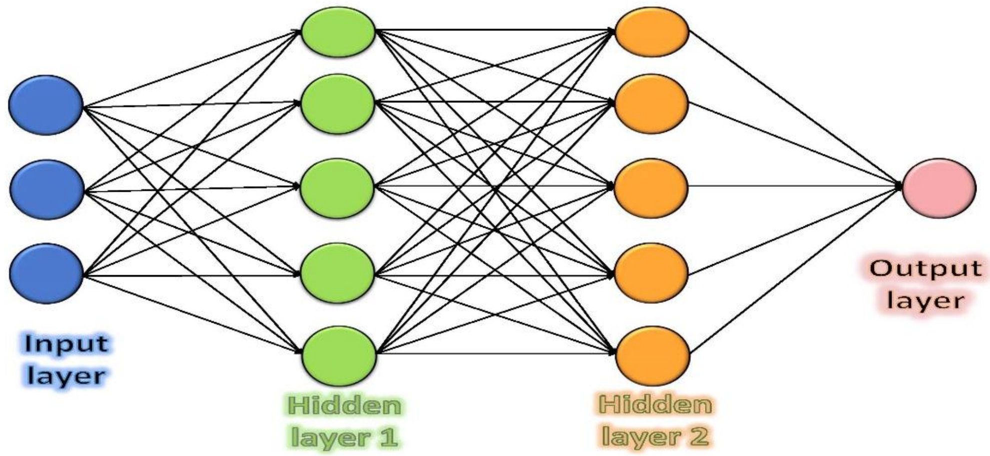


Figure 2.1. General feed-forward neural network (FNN).

2.4. Recurrent Neural Networks (RNNs)

FNNs are enhanced in Recurrent Neural Networks (RNNs) by including cycles, allowing them to 'remember' previous inputs. Memory is especially beneficial for time series forecasting as an inherent property, especially for pregnancy prediction, as shown in (Figure 2.2) simple recurrent neural networks where past load values directly impact future load values[35]. When trained for temporal data, RNNs effectively capture temporal dependencies associated with load forecasting problems [36]. Nevertheless, basic RNNs may produce inaccurate results when dealing with long-term dependencies based on problems such as vanishing gradient, because of which further enhancements to the RNN architecture have been made, such as LSTM network; they have been widely integrated into load forecasting problems recently [6], [37].

2.5. Convolutional Neural Networks (CNNs)

Artificial Neural Networks (ANNs), such as Convolutional Neural Networks (CNNs), have been used in load forecasting despite their original application in image processing, especially where big data sets can be processed without involving hand-crafted feature extraction. CNNs are effective at capturing the spatial-level structure present in data, and these features can be transferred to the temporal structure of time-series data in the case of CNNs [17]. In load forecasting, CNNs have been adopted to identify complex relations in the data as spatial structures and temporal sequences in one net and often combined with RNNs [38]. As discussed earlier, CNNs are beneficial in tasks that involve data of a grid-like structure or in cases whereby the load data includes data from different periods [39], as shown in (Figure 2.3.).

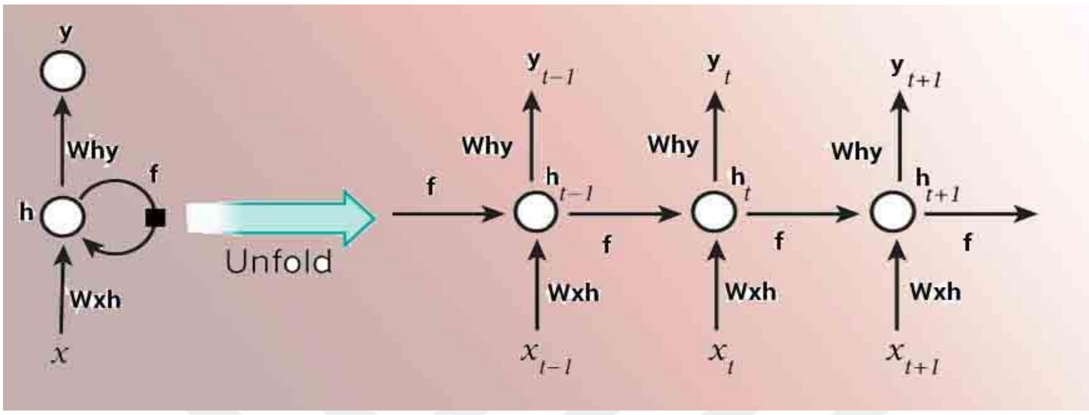


Figure 2.2. Simple RNN architecture.

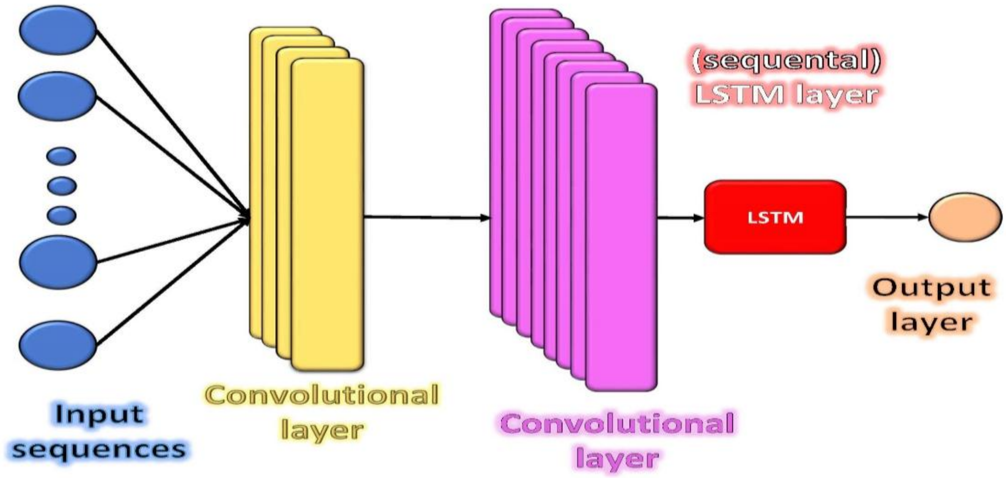


Figure 2.3. CNN-LSTM-forecasting model architecture.

Given these advantages, researchers & practitioners could get more accurate and more robust load forecasts by appropriately applying these neural networks and selecting their architecture according to the characteristics of the data and the forecast horizon.

2.6. Feature Selection and Data Preprocessing

The feature selection and preprocessing of the data is an important preparation to ensure that forecasting for loads is accurate and robust, especially for the models that use neural networks. These steps help ensure that only good, relevant and less noisy data is fed into the model so that the model can learn from the data.

2.6.1. Feature selection

Feature selection is the process of choosing only those variables or features from the dataset that contribute significantly to predicting the target variable, which is electrical load. Feature selection helps reduce the model's complexity by focusing on the most influential features. It also reaps several other benefits, which are highlighted below. The feature selection reduces the overall time taken by the model to train. Therefore, upon achieving the best results, the time required to train the model is significantly reduced [40]. Feature selection can be done in different ways. One is through statistics, whereby features correlated to one another and highly correlated with the target variable are selected [41]. Other techniques, including mutual information and recursive feature elimination, have also been applied effectively in load forecasting whereby the model can identify non-linear relationships between given features and load [42],[43].

Feature selection is very important in load forecasting because the input data can be almost anything, from load history to weather conditions, economic data, and even demographic data. Feature selection enhances model performance and is an anti-overfitting approach when dealing with big data sets [44], [45].

2.6.2. Data preprocessing

Data preprocessing is the pre-processing step that prepares data for the modeling phase. This step is crucial in load forecasting because raw data is not necessarily clean, which may consist of noise, missing values and outliers, which will adversely affect the performance of the neural networks [46]. Some regular preprocessing methods include data normalization whereby the pattern data is normalized to a fixed range, either from 0 to 1 or -1 and 1, respectively, to make the features comparable and hasten the neural network's learning process [47].

The second step of data preprocessing is cleaning, where missing data and other errors must be solved. Depending on the nature and the severity of the problems with data, one can use techniques such as interpolation, imputation, or even just the exclusion of problematic cases [48]. Also, techniques such as data generation, where new data points are created to complement existing ones, can enrich the data set, especially when the data is scarce or unbalanced [49].

Other parts of data preprocessing include temporal alignment or aligning features in time when joining the data with other data, such as weather data and past load data [50]. This alignment is crucial for accurate load forecasting because mismatched data can sometimes make wrong predictions or even reduce the model's performance.

Therefore, feature selection and feature preprocessing should be the critical and essential steps in applying neural networks to load forecasting. These steps guarantee that the model is trained in quality and relevant data, hence the right predictions.

3. RESEARCH METHODOLOGY

To accurately predict the electrical load considering the factors affecting the load, from different climate and human changes, there is no doubt that accurate prediction must depend on a general approach that explains the behavior of the work; in this research, work has been done on several main points that have been clarified in the block diagram (Figure 3.1). This section will cover the various methods that led to the implementation of a neural network using TensorFlow as well as the use of many modern techniques, including Normalization, Dropout Layers, Adam Optimizer and the use of Early Stopping techniques, which led to playing major and decisive role in the accuracy of load prediction.

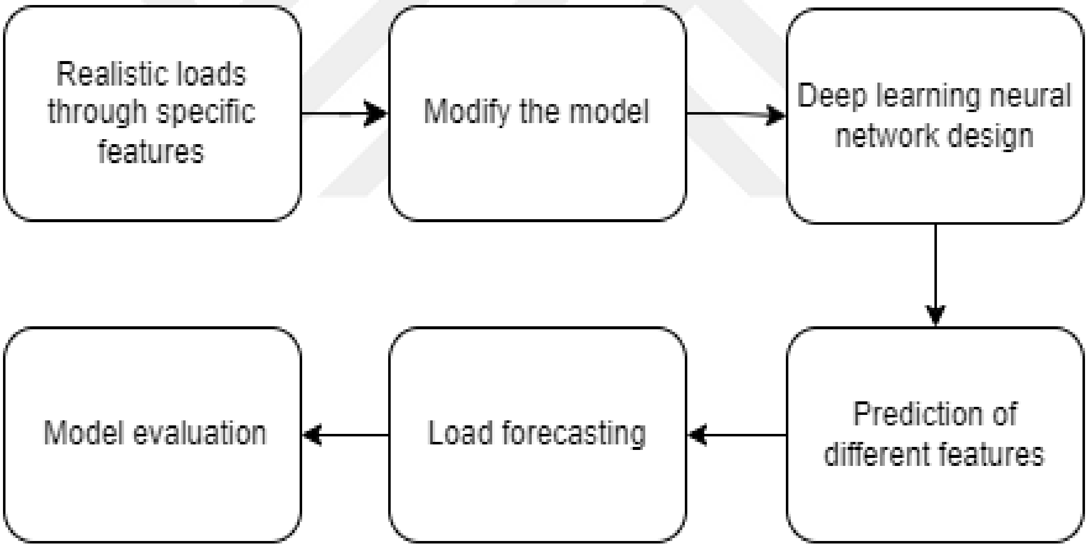


Figure 3.1. The block diagram approaches predicting the electrical load.

3.1. Data Collection and Data Sources

3.1.1. Historical data from electricity companies

Historical data on electrical loads for Babil Governorate were collected from the General Company for the Distribution of Middle Euphrates Electricity, especially from electrical power distribution stations (Figure 3.2.).

Figure 3.2. Substation 33/11kV for electricity distribution [55].

Table 3.1 shows the number of distribution substations 33/11k.v according to the years and the increase that occurred to accommodate the growing electrical loads.

Table 3.1. Total distribution substations feeding 11 KV lines.

Years	Number of Fixed Substations	Number of Mobile Substations	Total Distribution Substations Feeding 11 KV Lines	Number of Annual Added Substations
2020	41	12	53	
2021	42	15	57	4
2022	47	16	63	6
2023	52	19	71	8
2024	54	19	73	2

The graph below shows the increase in the number of 33/11 kV distribution stations and 11 kV feeders in Babel Governorate for the years 2021 to 2024.

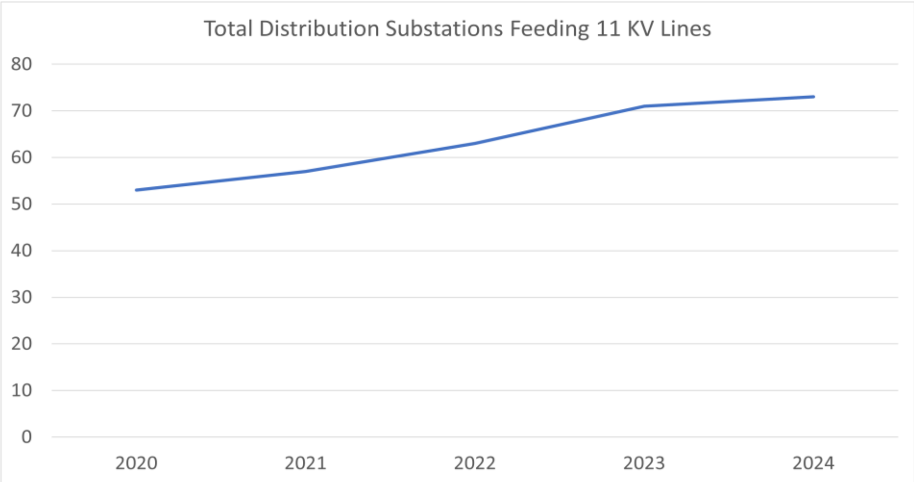


Figure 3.3. Total distribution substations.

Also, in Table 3.2, the number of 11 kV feeders from which loads were taken over the years 2021-2024 was measured, and the graph in Figure 3.4 shows the amount of growth in the 11 kV line network.

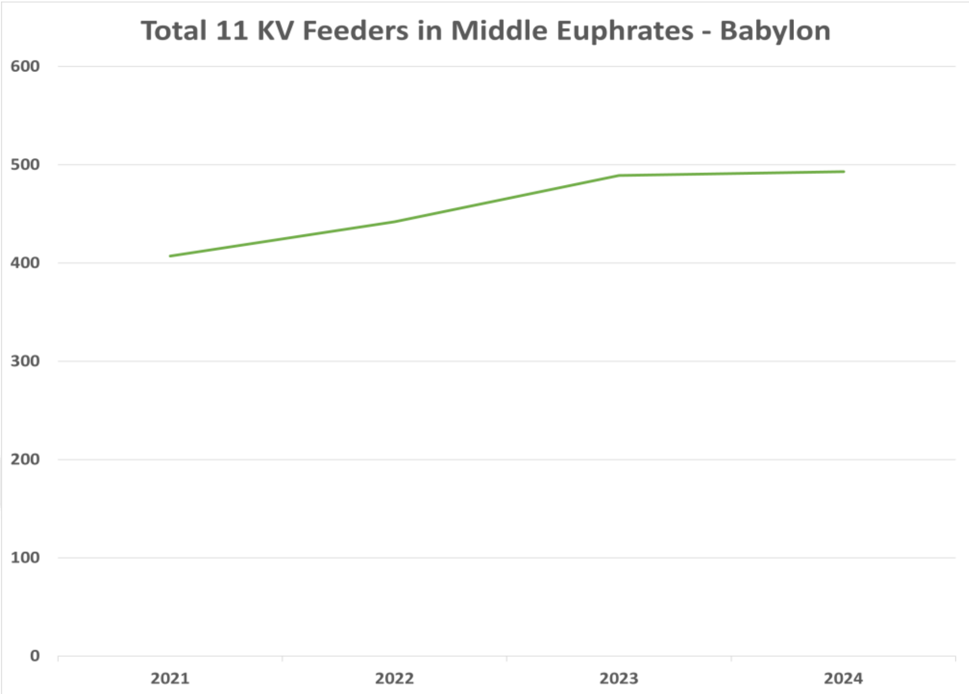


Figure 3.4. A total of 11 KV feeders in the middle euphrates – babylon.

Table 3.2. A total of 11 KV feeders in the middle euphrates – babylon.

Years	Total 11 KV Feeders	The Amount of Annual Increase
2021	407	21
2022	442	35
2023	489	47
2024-March	493	4

These substations include data and readings of electrical loads for 33 kV and 11 kV feeders appearing on 33 kV and 11 kV circuit breakers and installed with electrometers, as in Figure 3.5.

Figure 3.5. 33 kV & 11 kV Circuit breakers [56].

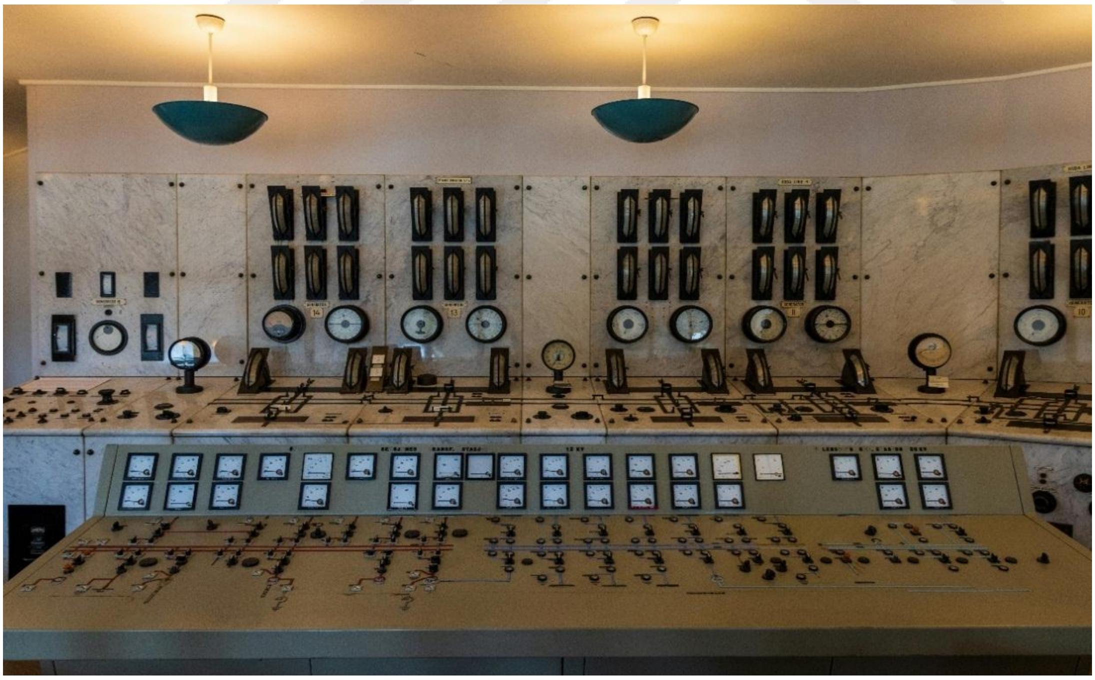


Figure 3.6. Electrometers to read loads, power factors, voltage, and current [57].

As these meters in Figure 3.6 show various readings of voltages, currents, watts, etc., The various readings of the distribution stations spread throughout the governorate were collected in Table 3.3, which shows the monthly electrical loads from January 2021 to March 2024**.**

3.1.2. Climate data available from the public domain

Temperature data were obtained from high-accuracy and standard worldwide websites like AccuWeather, Wunderground, and Weather. In the same period, weather data contain max/min temperatures, which have a massive impact on electricity usage (see Table 3.3).

3.1.3. Economic and social data

During those events, industrial and commercial activities and religious events such as Ashora and Arbain ceremonies greatly impact electricity consumption in the governorate.

Table 3.3. Electrical loads and temperatures for previous months.

Years	Months	Highest load Registered (MW)	Temperature Great	Temperature Minor
2021	January	1,908	16	6
2021	February	1,606	18	8
2021	March	1,228	22	11
2021	April	1,080	27	16
2021	May	1,717	34	21
2021	June	2,013	42	27
2021	July	2,076	45	29
2021	August	2,197	46	30
2021	September	2,008	41	26
2021	October	1,407	31	21
2021	November	1,205	24	14
2021	December	1,420	17	7
2022	January	1,942	18	7
2022	February	1,720	19	9
2022	March	1,340	23	12
2022	April	1,158	28	17
2022	May	1,879	35	22
2022	June	2,159	43	28
2022	July	2,235	46	30
2022	August	2,321	47	31
2022	September	2,137	42	27
2022	October	1,586	32	22
2022	November	1,364	25	15
2022	December	1,632	18	8
2023	January	1976	18	7

Table 3.3. (Continued) Electrical loads and temperatures for previous months.

Years	Months	Highest load Registered (MW)	Temperature Great	Temperature Minor
2023	February	1851	20	10
2023	March	1458	24	13
2023	April	1429	29	18
2023	May	2041	36	23
2023	June	2,308	44	29
2023	July	2,352	47	31
2023	August	2,395	48	31
2023	September	2,254	43	28
2023	October	1,726	33	23
2023	November	1,557	26	16
2023	December	1,820	19	9
2024	January	2,047	18	7
2024	February	2,034	20	9
2024	March	1,629	24	13

3.1.4. Websites and government databases

Information on the gross domestic product, the population census regarding the governorate of Iraq and the per capita income of Iraqis were obtained from government agency websites, including the Central Statistical Organization.

3.2. Data Preprocessing Techniques

3.2.1. Data cleaning

The data processing stage is an important step in data preparation for modeling. This stage includes several steps: This stage includes several steps:

- Removing missing values: This might entail processes such as interpolation, which replaces the missing values with the average of the neighboring values.
- Dealing with outliers: Outliers that might influence the outcome detected and either removed or altered.
- Data Formatting: They involve converting the data into a common format to enhance its analysis since it has been tested and is efficient.

3.2.2. Techniques used in the preparation of data transformation and modeling

It is further transformed into the right format from the data cleaning steps in the investigation stage. These steps include:

- Normalization: Another application of feature scaling methods is using algorithms like Min-Max Scaler to rearrange the features to fall within a range of small or large values, say between 0 and 1. Which in turn enhances the efficiency of the machine learning models. The following equation is used:

$$\mathrm{X\ scaled} = \frac{\mathrm{X} - \mathrm{X\ min}}{\mathrm{X_{max}} - \mathrm{X\ min}} \tag{3.1}$$

  Where (X) represents the original feature value, (Xmin) and (Xmax) are the lower and upper limits of the feature.

- Data Splitting: Dividing the dataset into training and testing datasets to prove their generality and prevent the data from being oversampled.
- Identifying Independent and Dependent Variables: Deciding between independent variables, which include temperature, and dependent variables, which include electrical loads, before adapting them to the modeling.
- Data Resampling: The data is reshaped to the correct dimension using data sequence-based models such as LSTM.

3.2.3. Feature engineering

The feature engineering process is concerned with filtering and transforming the given raw data into the feature space that is more suitable for the problem at hand. TEMP" For this study, the features selected are "Temp. Great" and "Temp. Minor" refers to a large temperature differential that influences the electrical load.

3.3. Neural Network Design

Neural network architecture defines the structure of the model, including layers, the number of neurons, and even active functions and other features.

3.3.1. Types of layers

- Dense Layers: It is important to note that dense layers are the primary layers of neural networks. Every neuron in the current layer interacts or relates to each neuron in the next layer. If the layers are learned in the right order, these fourteen layers can allow the model to learn the right patterns of the input data.
- Dropout Layers: Cropped layers are also applied in practice to minimize overfitting risks, which occur during the training part of the neurons that are switched off randomly. This is useful when reducing overfitting to improve the model's capacity to learn from other new data sets. This neural network is designed using a sequential model in TensorFlow (see Figure 3.7.) with the following architecture:

  Input Layer:

  Dense Layer (128 Neurons, ReLU Activation)

  Dropout (0.2)

  Hidden Layers:

  First Hidden Layer:

  Dense Layer (64 Neurons, ReLU Activation)

  Dropout (0.2)

  Second Hidden Layer:

  Dense Layer (32 Neurons, ReLU Activation)

  Dropout (0.2)

  Output Layer:

  Dense Layer (1 Neuron, Linear Activation)

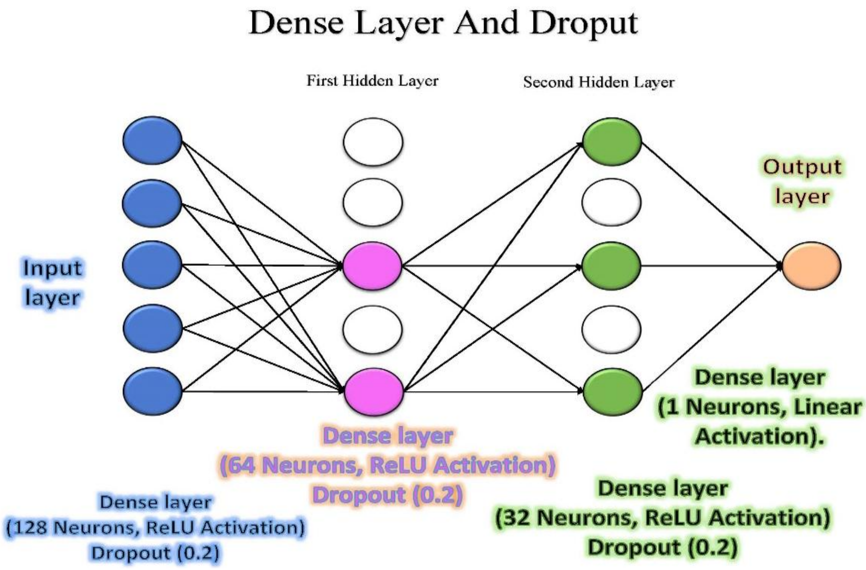


Figure 3.7. The neural network architecture in tensorflow.

- LSTM Layers (Long Short-Term Memory): LSTM layers are employed where the data is temporal or sequential since they can handle such data efficiently (Figure 3.8.) because they provide the ability to keep some information for an extended time.

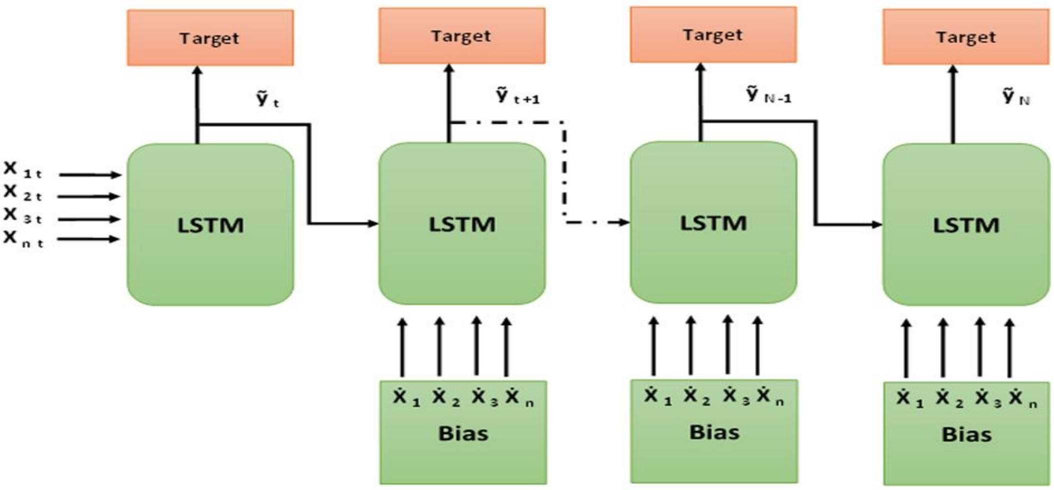


Figure 3.8. Long-short-term memory (LSTM) network model overview.

3.3.2. Number of neurons

- Layer 1: This layer mainly comprises many neurons (for example, 128 neurons of this layer) because this layer takes the raw data and then begins extracting significant features.

- Layer 2: This layer has fewer neurons, for instance, 64 neurons, because this layer starts learning more complicated features extracted from the first layer.
- Layer 3: This layer contains fewer neurons (for example, 32 neurons), decreasing complexity gradually and paying more attention to features.

In line with this, as shown in Figure 3.7, is the number of neurons and layers.

3.3.3. Activation functions

- ReLU (Rectified Linear Unit) function: Why the ReLU function is applied in neural networks. The ReLU function is used in neural networks because it resolves the vanishing gradient dilemma and enhances the training pace.
- Linear Activation function: This is applied in the final outputs to solve electrical load prediction problems where you expect a continuous value as the output variable (electrical load). Figure 3.9. shows linear activation function and Figure 3.10. activation function nonlinear.

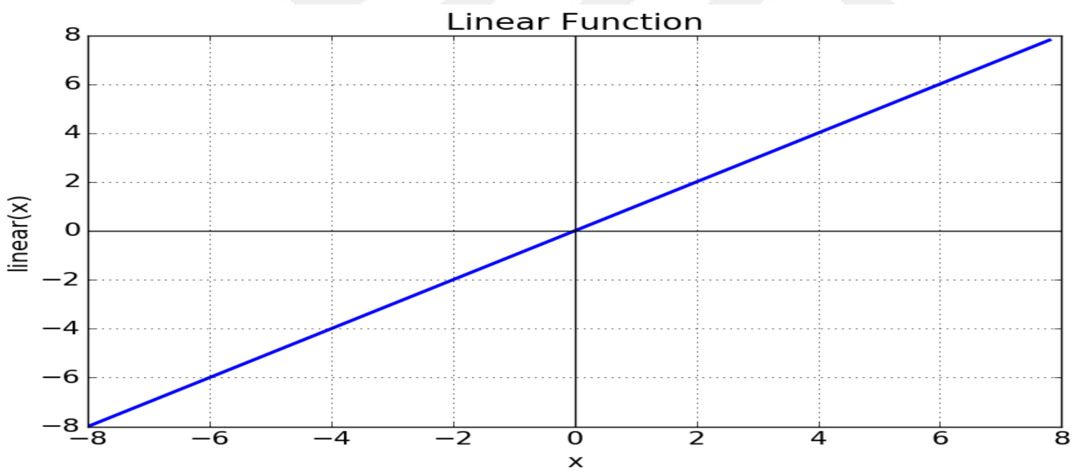


Figure 3.9. Activation functions in neural networks-linear function.

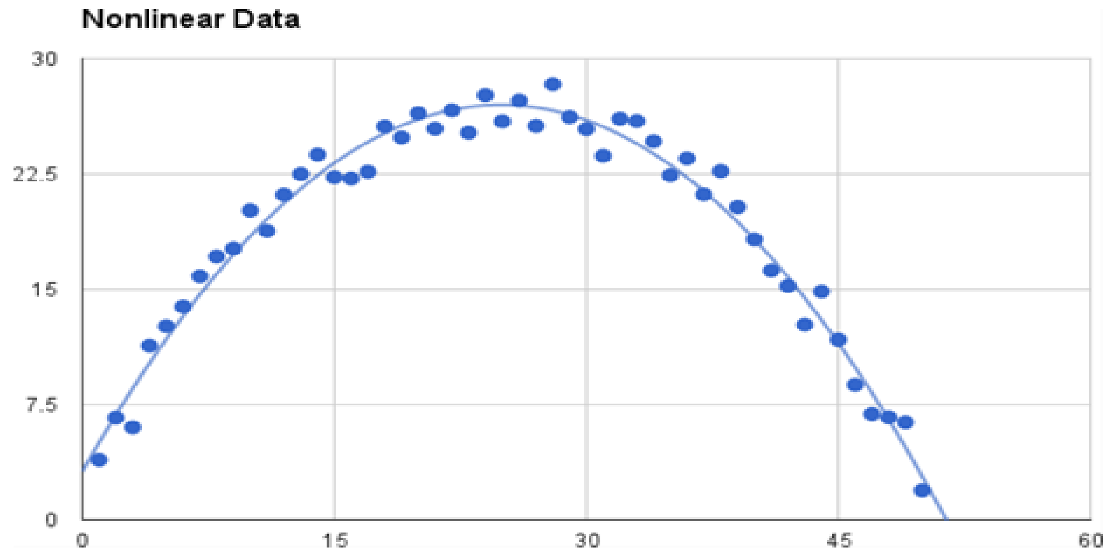


Figure 3.10. Nonlinear activation function.

3.3.4. Hyperparameter selection

Key hyperparameters for the model include:

Learning Rate: The parameter was set at 0.001 at the beginning. And the generator G is trained by GAN loss Cross Entropy with a learning rate scheduler.

Batch Size: 32

Epochs: 500. To avoid over-fitting, the early stopping was done.

3.4. Model Training and Optimization

Training and Optimization Procedures: The model is trained using Adam optimizer (Mean Squared Error) MSE as the loss function, and other tricks are also applied, which are discussed below.

3.4.1. Adam optimizer

Adam optimizer is used to optimize the weight during the training process due to its efficiency in handling large datasets.

$$\theta_{(1+xt)} = \theta_t - \frac{\eta}{\epsilon + \sqrt{v_t}} m_t \tag{3.2}$$

Where

m_t refers to an estimate of the first moment.

ν_t an estimate of the second moment.

η is the learning rate.

ϵ is a very small number used to avoid division by zero.

3.4.2. The first candidate is the mean squared error (MSE) loss function

MSE is among the widely used loss functions in machine learning, particularly in regression problems. This function compares the actual values obtained with the values the model gives during the learning process in the training phase (Figure 3.11).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_t - \hat{y_t}\right)^2 \tag{3.3}$$

Where:

(y_t) is said to be the actual value.

$(\hat{y_t})$ This is represented Predicted value of system.

(n) is the number of observations available in the data set.

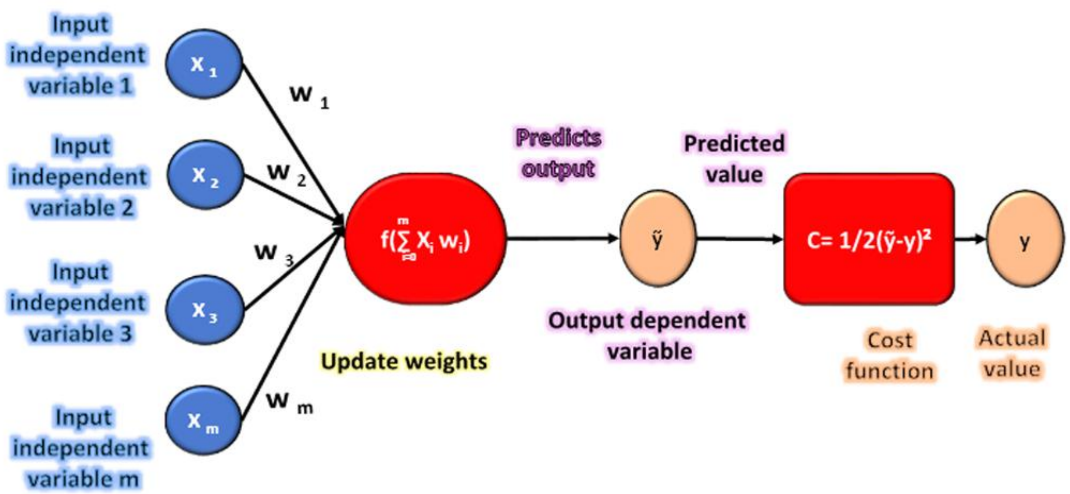


Figure 3.11. Loss functions for neural networks.

3.4.3. Learning rate scheduler

The learning rate is reduced after 10 epochs:

$$\eta_{\text{new}} = \eta_{\text{initial}} \times e^{-0.1} \tag{3.4}$$

3.4.4. Hyperparameter tuning

Hypers parameter tuning may, therefore, be described as fine-tuning or altering the crucial parameters of the model to enhance its functionality. This can be achieved in several ways:

- Grid search: The grid of possible values is constructed for each parameter, and the model is trained on every combination of the values to find out which set is the best.
- Random search: Each parameter is assigned a random value within a specified range, and all these values are tested until the best is obtained.
- Bayesian optimization: Random selection is utilized to find the parameter sets with the least number of tries through probabilistic techniques.

3.4.5. Dropout layers

This is quite a typical problem in the context of deep learning models where the model captures details and noises from the training set, which will not yield the desired results when applied to another set. To avoid this challenge and reduce the chances of overfitting, we use dropout layers, whereby these layers neglect some of the neurons during training. This helps the neurons not to be too reliant on one another, contributing to an increased procedural recognition in the model.

3.4.6. Regularization techniques

Overfitting is prevented by regularization, which is done using projection layers. During training, each projection layer of the input units is set to zero randomly, a portion of the inputs for every update, which minimizes the co-adaptation of hidden units. This makes the avoidance of co-adaptation of hidden units possible. Other regularization terms can be incorporated into the loss function, such as L1 and L2 regularization, often called quadratic and absolute regression. This works against large values of weights and nudges the model in the right direction to learn easily.

3.5. Performance Evaluation

The model's performance is evaluated using the following metrics:

3.5.1. Mean absolute percentage error (MAPE)

Mean Absolute Percentage Error: a measure in performance assessing accuracy calculated in actual figures. MAPE can be advantageous in a model comparison evaluation because it has been harvested from datasets with varying data scales [51]. Let's talk about MAPE usage in load forecasting. It becomes especially helpful when we look only at a relative error of forecasted values concerning the actual load rather than the absolute error. However, Hyndman and Koehler also highlighted the disadvantage of MAPE: it leads to large percentage errors when actual values are very small [52].

$$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y_t}}{y_t}\right| \tag{3.5}$$

Where:

(y_t) is said to be the actual value.

$(\hat{y_t})$ This is represented Predicted value of system.

(n) is the number of observations available in the data set.

3.5.2. Mean absolute error (MAE)

The Mean Absolute Error (MAE) was used to evaluate the models' performances, which is the mean of the absolute differences between actual and predicted values. Besides, it offers a simple and easy-to-measure way of determining the accuracy of each model.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y_i}\right| \tag{3.6}$$

Where:

yi are the actual values collected from the real-life situation.

$y\hat{}i$ are the predicted values.

n is the number of observations available in the data set.

3.5.3. R-Squared (R²)

As it's widely known, the Coefficient of Determination is usually represented by R^2. It ascertains the percentage of dependent variable variability that the independent variables can reasonably explain. Values of R-squared range from 0 to 1, with the best values positioning the model as the best surrogate for data. The model perfectly described the data. The coefficient of determination signifies no structural dependency at zero [53].

$$\mathcal{R}^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.7}$$

Where:

yi are the values used to observe the condition and calculate the results.

yy^i are the mean values where statistic models have been used or predicted values where machine learning algorithms have been applied.

$y\bar{}$ is assumed to be the mean of the actual values of the observation.

3.5.4. Explained variance score

Explained Variance Score is one of the ways of assessing the probability of the predictions made by the model, whereby the percentage of variance in the predictions is determined. It gives us information about the model's fitness to estimate the variability of the target variable.

$$\text{Explained Variance Score} = 1 - \frac{Var(y - y\hat{})}{Var(y)} \tag{3.8}$$

Where:

Var (y −y^) is the variance of prediction errors.

Var (y): It represents the variance by actual value.

3.5.5. Cross-validation

It is to be noted that most of the steps in the code do not crudely relate to cross-validation. However, each data set is divided between training data and testing data through an 80:20 division to make sure that the model is tested on data that the model has not seen any data set is divided between training data set and testing data set through 80:20 division which acts as a very rudimentary form of validation.

3.6. Model Implementation

3.6.1. Implementation details

After the data had been gathered, sorted and evaluated, they were divided into training and testing data sets. A neural network model was built using TensorFlow, new data features were reshaped, and data was predicted in the process during which the model of the program was also evaluated. The mechanism of program implementation is presented in Flowchart as described in Figure 3.12.

The following flowchart shows the program code implementation of the students' affairs section.

3.6.2. Tools and libraries

Using the programming language, we had Python 3. 9 / In Spider version 5; the models were done in TensorFlow and Keras in Python. TensorFlow allows for successful calculations using neural networks, and Keras allows for simplifying the model. With the help of pip Python's package installer, Pandas was used to read and clean data from CSV files, NumPy was used for efficient numerical computations and processing of the data, TensorFlow was used for building deep learning models as well as to train the models, Scikit-Learn was used to normalize the data and to split it into training and testing data and to calculate the various performance metrics. It also offered assessment and tuning instruments; while I did this, I improved the text formatting and printout from Rich.

3.6.3. Data loading

We import and load data where data is already available in CSV format and place them in a single data frame. It can have any information but probably contains temperature and electrical load information.

3.6.4. Data processing

Dealing with the missing and outlier data in the clean data set. Normalizing data using MinMaxScaler.

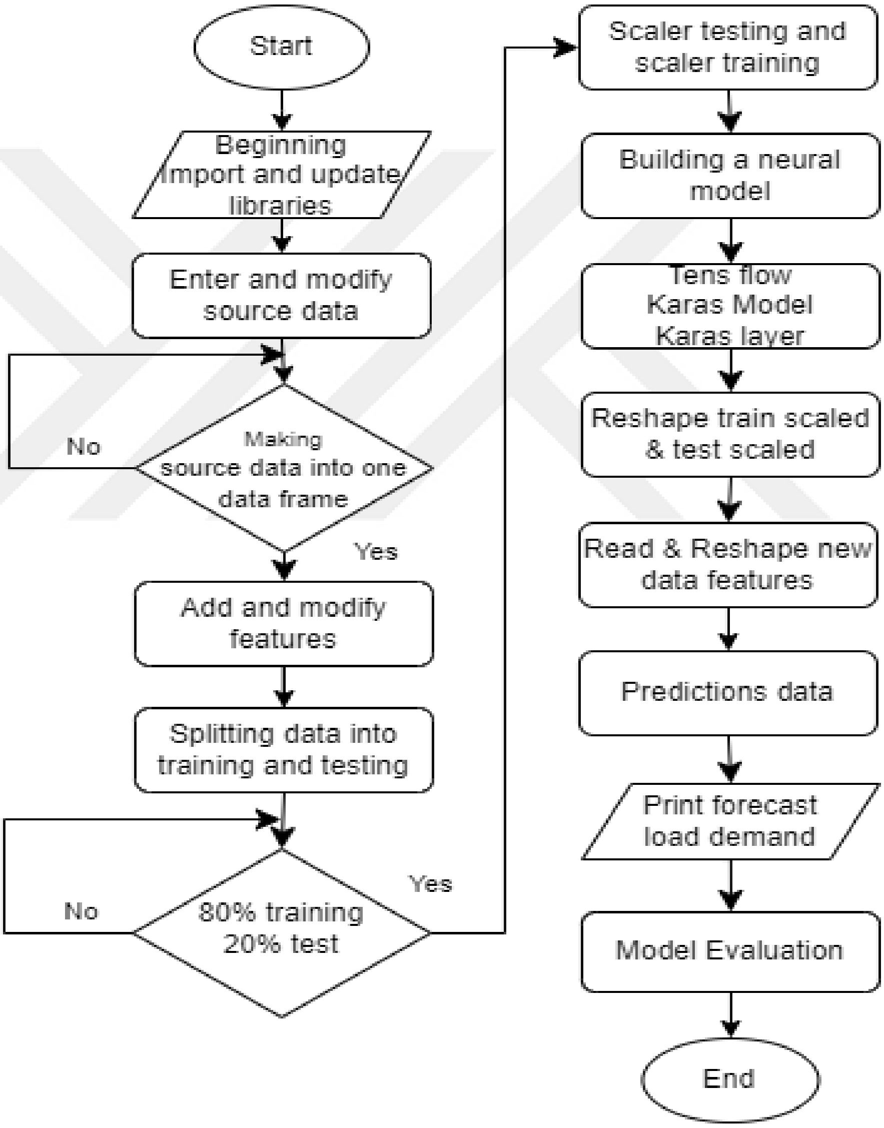


Figure 3.12. Flowchart of the program code implementation.

3.6.5. Data splitting

The data will be split into training and testing sets to use train_test_split. The training datasets and labels have divided the dataset into the training sets of individual labels by percentage to an 80/20 ratio. The features selected for training with the model were Temp. Great and Temp. Minor, while the target was to set demand values between 1000 and 3000.

3.6.6. Building the model

About building a deep neural network model with the help of TensorFlow library. The model can also contain other hidden layers, such as dense layers with ReLU activation and dropout layers, where these layers are placed between the hidden layers to prevent them from overfitting. Network initialization was performed with the ReLU activation function for nonlinearity, and the last layer used a linear activation function to provide a continuous load demand prediction.

3.6.7. Training the model

The training data will be used to train the model, and then the number of iterations will be stated. Employing the concept of early stopping to overcome/ prevent the problem of overfitting. The training was done on a local machine with a GPU for computational purposes and was used. The presented model was trained for 500 epochs. However, early stopping was used during the training to avoid overfitting the model, which can be switched on in case no improvement in validation loss is detected in the course of fifteen epochs.

3.6.8. Optimization and tuning

Several other hyperparameters, such as the presence of learning rate, batch size and the number of neurons, were adjusted based on the approach known as grid search. The compression rate scheduler alters the learning rate depending on the current epoch and the flocking goal of epochs, and the learning rate slowly decays as the training progresses more accurately.

The equation used to adjust the learning rate is:

$$Ir_{new} = Ir_{initial} \times e^{-K.epoch} \tag{3.9}$$

Where:

lr $_{new}$ is the new learning rate used in the current epoch. The variable lr represents the specified learning rate for iteration epochs.

lr $_{initial}$ is the initial learning rate at significant start training operations were performed at this rate.

k is the decrement constant. The value of k determines the slope of the decrement in activity level from its initial value a.

Epoch is the current epoch number. Conventionally, the epoch measures time, and this is the current epoch number.

e is the mathematical constant of the natural logarithm base (Euler's number, approximately equal to 2.718).

Model Optimization Techniques Adam optimizer was used as a learning algorithm with adaptive learning rate properties that converge faster than SGD, RMSProp, AdaDelta and AdaGrad. Dropout layers and regularization methods were incorporated into the network design to tackle the overfitting problem.

3.6.9. Model evaluation

Evaluate the performance of the model using test data. Calculate metrics such as MAPE, MAE, R2 and Explained Variance Score to evaluate the model's accuracy.

4. DATASET AND MODELS

4.1. Dataset Properties

The dataset used in the study comprised a large sample of 17,150 values for 11 kV feeder loads and maximum and minimum temperatures from January 2021 to March 2024 (39 months), providing a comprehensive representation of seasonal and temporal fluctuations. However, the data exhibited a degree of variability and irregularity due to various factors such as seasonal variations, exceptional events, and religious pilgrimages. This necessitated applying advanced artificial intelligence and machine learning techniques to ensure accurate load forecasting. Table 4.1 represents the total dataset and the training, validation, and testing samples.

Table 4.1. Total dataset, training, validation, and testing samples

Dataset samples	Training samples	Validation samples	Testing samples
17150	8654	2164	6332

4.2. Applied Models

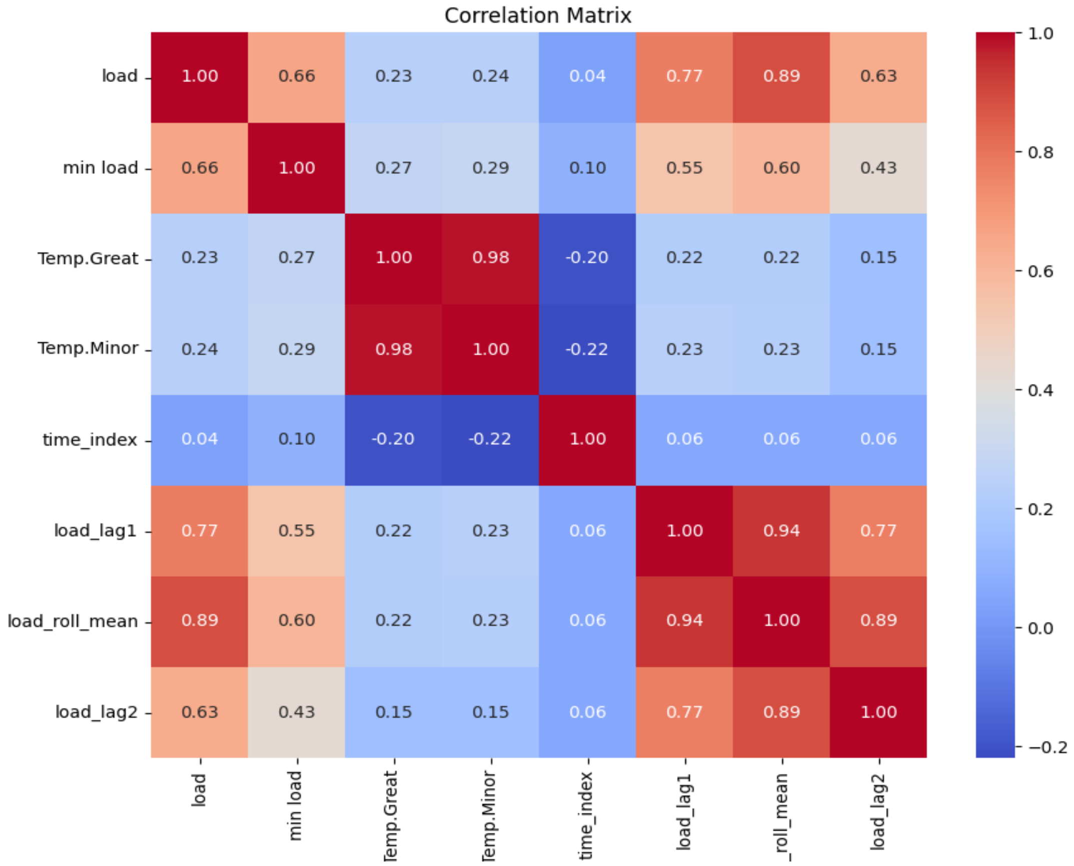
4.2.1. Forward neural network model (or a multilayer perceptron, MLP)

We have already learned about feed-forward networks in Section 2.3, and now we will learn about them as a model used in the program code and the most important aspects of the main processing that will be highlighted in this study:

- Temporal Feature Handling:
  - Month-to-numeric conversion
  - Continuous time index creation
  - Cyclic encoding for seasonal patterns
- Advanced Feature Engineering:
  - Multiple lag features (lag1, lag2)
  - Rolling window statistics
  - Temperature-derived features (range, average)

- Data Normalization:
  - Min-max scaling for all numerical features
  - Consistent scaling across train/validation sets
- Sequence Preparation:
  - Per-feeder sequence generation
  - Fixed-length time steps (9 in this case)
  - Proper temporal ordering preservation
- Validation Approach:
  - Time-based split (more realistic than random)
  - Consistent feeder representation in both sets
- Quality Control:
  - Comprehensive NaN handling
  - Correlation analysis as shown in the correlation table
  - Shape verification at each step

Table 4.2. The correlation matrix.



This model demonstrates a robust methodology for preparing time-series load forecasting data, addressing key challenges such as temporal dependencies, multiple feeders, and feature scaling. The approach maintains the data's temporal integrity while creating features that capture both short-term and seasonal patterns in electricity consumption; Table 4.2 shows a correlation matrix of the model.

The block diagram in Figure 4.1 illustrates the architecture mechanism work of a deep neural network model that uses two different input branches. The first branch processes sequential input data (sequence_input) with a shape of (9, 8), passing it through several Dense, Flatten, and Dropout layers. The second branch processes an additional input (feeder_input) through an Embedding layer followed by Flattening. The outputs of both branches are then merged in a Concatenate layer, followed by additional Dense layers to produce the final output.

This design enables the model to combine sequential information with categorical or numerical data in a unified architecture.

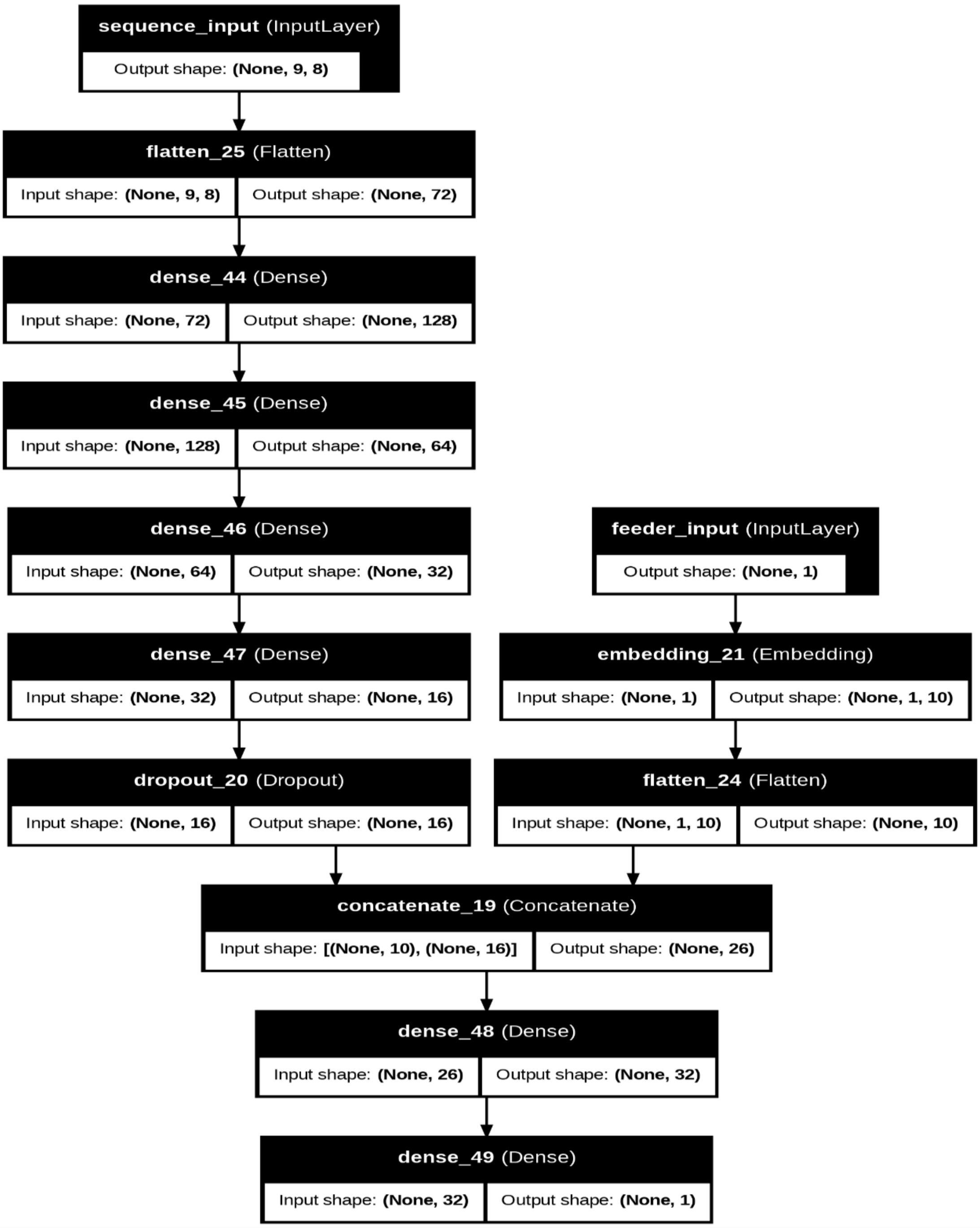


Figure 4.1. Block diagram of the FNN model.

4.2.1.1. Model summary

Table 4.3 summarizes the FNN model, a summary of a deep neural network model built using Keras Sequential API. The model consists of several Dense (fully connected) layers with 128, 64, and 32 units, each followed by a Dropout layer to prevent overfitting. The final output layer has a single unit suitable for regression or binary classification tasks. The table displays the output shape and the number of trainable parameters for each layer in the model.

Table 4.3. Model summary of FNN.

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 128)	1,152
dropout_6 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 64)	8,256
dropout_7 (Dropout)	(None, 64)	0
dense_10 (Dense)	(None, 32)	2,080
dropout_8 (Dropout)	(None, 32)	0
dense_11 (Dense)	(None, 1)	33

4.2.1.2. Training and validation loss

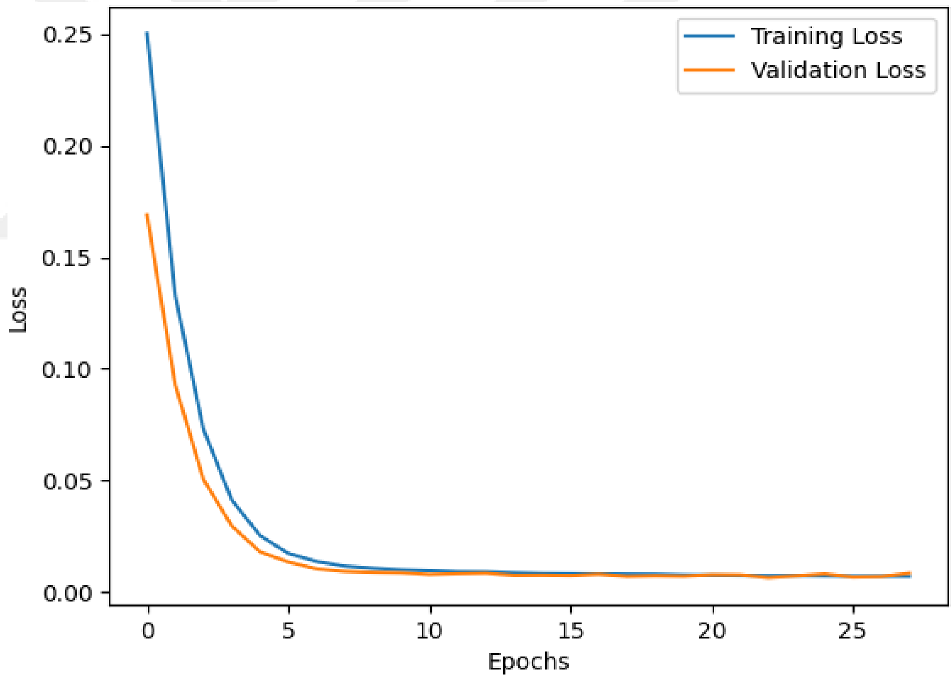


Figure 4.2. Training loss and validation loss of the FNN model.

Analysis of the training loss and validation loss curves showed that the model achieved efficient learning over the training epochs, with the training loss steadily decreasing to ~0.05 at epoch 25, indicating the model's ability to learn from training data with high accuracy (see Figure 4.2).

4.2.1.3. Evaluation

Table 4.4. shows the summary of the FNN model, the MAE evaluation on the normal and original scales, and the percentage error.

Table 4.4. MAE evaluation table for the FNN model.

Model	MAE (Normalized Scale)	MAE (Original Scale)	Percentage Error (%)
FNN	0.0744	48.11	6.34

4.2.2. LSTM model

Also, we have learned about LSTM networks in Section 3.3.1, and now we will learn about them as a model used in the program code and the most important aspects of the main processing that will be highlighted in this study:

4.2.2.1. Data preprocessing pipeline

- Initial Data Cleaning Irrelevant Columns: Non-predictive fields (e.g., NO., max load) were excluded.
- Missing Values: Handled via feeder-specific forward/backward filling (ffill/bfill) to preserve temporal integrity, followed by dropna() for residual nulls.
- Feature Engineering: Categorical Encoding: Feeders were numerically encoded using LabelEncoder().
- Temporal Features: Months were mapped to integers (January=1,..., December=12).
- A synthetic time_index was created to represent chronological order: (df['years'] - df['years'].min()) * 12 + df['month'].
- Lag Features:load_lag1,load_lag2:Previous timestep values (autoregressive properties). load_roll_mean: 3-step rolling mean (short-term trend capture).
- Normalization: Applied MinMaxScaler() to numerical features (load, temperatures, lags, etc.) to constrain values to [0, 1].
- Feature Selection Rationale

- Autocorrelation Analysis: Lag features were justified by partial autocorrelation (PACF) plots, indicating significant dependence on recent.
- Domain Knowledge: Temperature variables (Temp.Great, Temp.Minor) were included based on established correlations with energy demand.

The block diagram in Figure 4.3. illustrates the working mechanism of the LSTM model.

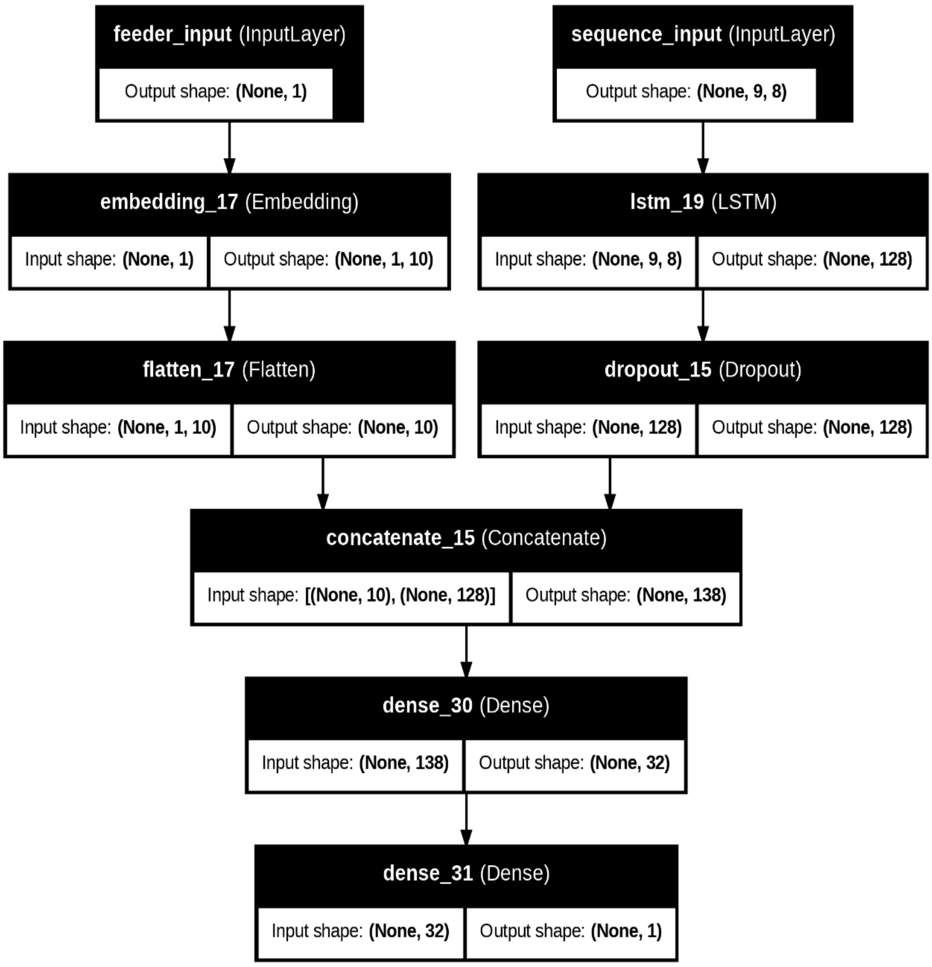


Figure 4.3. Block diagram of the LSTM model.

This diagram shows the architecture of a hybrid deep learning model that combines two input branches (see Figure 4.3). The first branch processes sequential data using an LSTM layer, while the second branch processes an additional input through an Embedding layer. The outputs from both branches are concatenated and passed through Dense layers to produce the final prediction. This design allows the model to integrate time-series features with categorical or numerical information for improved predictive performance (see Figure 4.3).

4.2.2.2. LSTM model summary

Table 4.5. Model summary of LSTM

Layer (type)	Output Shape	Param #	Connected to
feeder_input (InputLayer)	(None, 1)	0	-
sequence_input (InputLayer)	(None, 9,8)	0	-
embedding_16 (Embedding)	(None, 1, 10)	4,750	feeder_input[0] [0]
1stm_18 (LSTM)	(None, 128)	70,144	sequence_input[0] [0]
flatten_16 (flatten)	(None, 10)	0	embedding_16[0] [0]
dropout_14 (Dropout)	(None, 128)	0	1stm_18[0] [0]
concatenate_14 (Concatenate)	(None, 138)	0	Flatten_16[0] [0], dropout_14[0] [0]
dense_28 (Dense)	(None, 32)	4,448	concatenate_14[0] [0]
dense_29 (Dense)	(None, 1)	33	dense_28[0] [0]

Table 4.5 summarizes the architecture of a hybrid deep-learning model built with Keras. The model has two input branches: one processes sequential data using an LSTM layer, and the other processes additional input through an Embedding layer. The outputs from both branches are flattened and concatenated, then passed through Dense layers to produce the final prediction. The table shows the output shape, the number of trainable parameters, and the connections between layers. This design effectively allows the model to combine time-series features with categorical or numerical information.

4.2.2.3. Training and validation loss

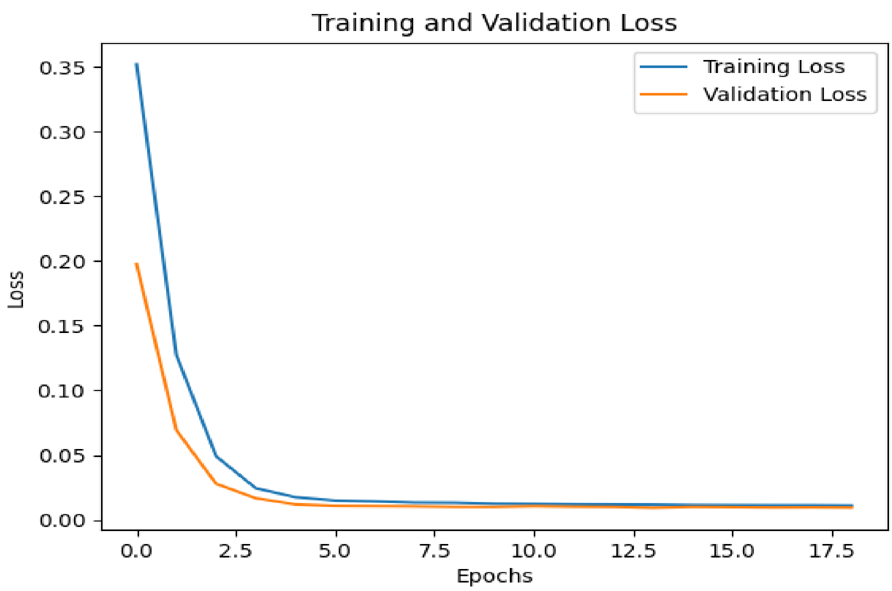


Figure 4.4. Training loss and validation loss of the LSTM model.

The training and validation loss curve analysis results showed clear stability in the model's performance, with the loss values for both curves decreasing simultaneously and steadily as the training epochs progressed. This parallel trend supports the model's ability to generalize well without showing signs of overfitting (see Figure 4.4).

4.2.2.4. MAE Evaluation was as follows

Table 4.6 summarizes the LSTM model, the MAE evaluation on the normal and original scales and the percentage error.

Table 4.6. MAE evaluation table for the LSTM model.

Model	MAE (Normalized Scale)	MAE (Original Scale)	Percentage Error (%)
LSTM	0.0566	36.61	4.83

5. RESULTS AND DISCUSSION

5.1. Exploratory Data Analysis (EDA)

The detailed analysis of the obtained data is called exploration data analysis, which helps to catch the tendencies of the dataset. It gives the first view of the data and aids in determining relationships between the variables and how they affect the prediction model. This research conducted exploratory data analysis of temperature and electrical load demand based on data collected from the Babil Governorate, Iraq.

5.2. Model Performance on Test Data

The electricity demand forecast for Babylon Governorate for the years 2024-2025 made by the neural network model reveals some fluctuations and trends related to temperature figures. The monthly arrivals are also associated with the dispersion of demand that, within the consecutive iterations, are represented more precisely by the model. The expected load demand values range from 1543 MW to 2334 MW from April 2024 to March 2025, as depicted in (Table 4.1) with the peak demand in the summer and winter months as these are the most preferred for cooling and heating, respectively. The set performance criteria suggest that the model produces excellent predictions regarding the mean square error in the test data (MSE), which reached approximately 95000, between (50,000 - 100,000) which are very good values indicating that the model can make accurate predictions. Also, the mean absolute relative error (MAPE) was 18%, less than 20%, which is good and completely acceptable considering the fluctuating nature of the loads in the period under measurement. These values and criteria explain the large temperature variation between summer and winter.

Table 5.1. Predicted electrical loads from april 2024 to march 2025.

Years	Months	load Forecast	Temperature Great	Temperature Minor
2024	April	1543	30	18
2024	May	1982	37	23
2024	June	2213	44	29
2024	July	2284	47	31
2024	August	2334	48	32
2024	September	2177	43	28
2024	October	1796	34	23
2024	November	1587	27	17
2024	December	1869	20	9
2025	January	2047	16	5
2025	February	2003	20	9
2025	March	1589	24	13

The foreseen results show that the model's potential for capturing seasonal periods and, consequently, the fluctuation of the period is fairly high, as actual and forecasted demand compares well, particularly during peak periods. It gives the results achieved about benchmarking data presented over similar periods of previous years, as depicted in Table 5.1.

5.3. Successes and Challenges

Enhanced accuracy of forecasts illustrates the predicted model's ability to capture temporal dependencies better than the feed-forward model. However, controlling the recursive layers and getting into vanishing gradients was a little difficult. Overfitting was another major problem solved with the help of Dropout, Early Stopping, and Gradient Clipping techniques.

5.4. Intervention of the Model Output

The model also estimated the peak load demand during the summer months, and the actual load estimated closely matched the forecasted value. However, variations of relativities were during transitional months that usually convey unpredictable demands; the marked example is in April and September, where slight disparities were detected (Figure 5.1).

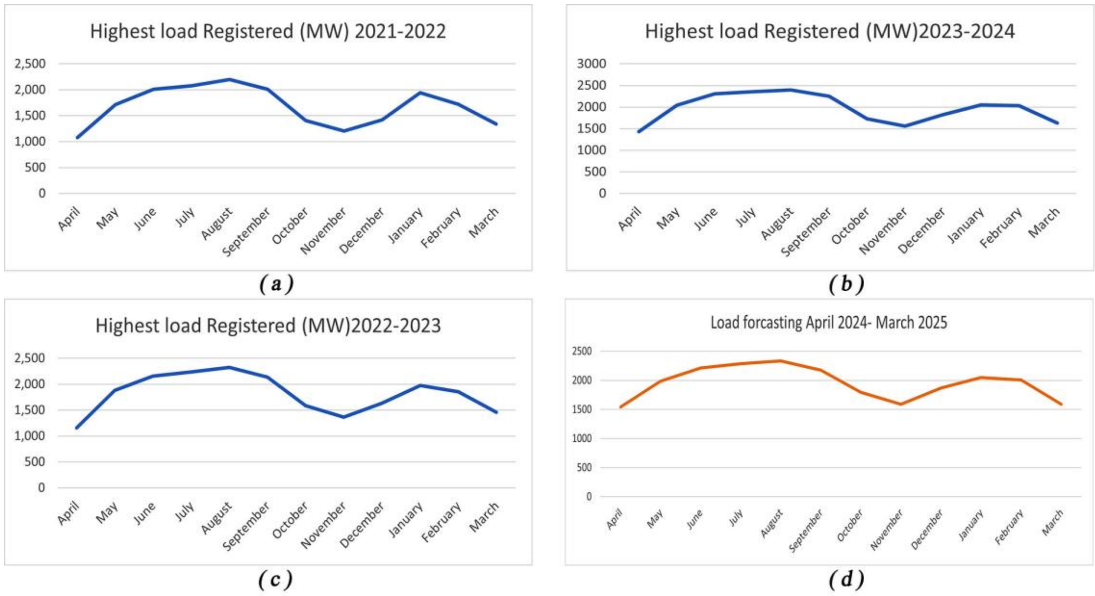


Figure 5.1. a) Highest load registered (MW) 2021-2022; b) Highest load registered (MW)2022-2023; c) Highest load registered (MW)2023-2024; d) Load forecasting April 2024- March 2025 - Comparison between expected and actual demand.

5.5. Effects of Varying Values

This study incorporated a sensitivity test to analyze the model's effectiveness regarding the various parameters. It was also discovered that batch size and learning rate played the most crucial role, where smaller batch size was confirmed to be more appropriate for training stability.

5.6. Model Robustness Analysis

Model stability was determined by checking the ability of the model to predict out-of-sample 2025 data. The model was very efficient when tested for future data sequences. Even though a slight deterioration in performance was detected, such areas were pointed out as areas that need further enhancement.

5.7. Discussion

5.7.1. Temporal and seasonal variations

According to the model performance, the load demand of each load at Babil Governorate illustrates the effect of temperature. As predicted, peak power consumption occurred in July and August as the temperature climbed to its peak [2]; it reached 48 °C and 51 °C in some of the months of this summer, and thus, all the essential facilities employed air conditioners in large quantities. Similarly, there is an increased demand in winter, particularly in January and February, since electric heaters are frequently used and temperatures range to 6 degrees Celsius. Hence, these results affirm the high sensitivity of electrical load to temperature changes, as reported in the load forecasting literature [10].

The model also predicts increased demand during holiday visits and religious occasions experienced by the governorate because of its closeness to the religious governorates of Karbala and Najaf.

April and November, which are considered moderate months, experienced low electricity consumption since heating and cooling were not so much required because of the moderate weather. This was due to relatively mild temperatures during these months, lowering total electricity consumption.

Overall, the accuracy of the demand was well predicted based on the model's forecasts. Still, a few issues needed improvement to capture the demand changes during transition periods, such as when unexpected demand was during the transitions from summer to winter or vice versa.

5.7.2. Comparative analysis of model performance

Table 5.2. Comparison between FNN and LSTM models

Model	MAE (Normalized Scale)	MAE (Original Scale)	Percentage Error (%)
FNN	0.0744	48.11	6.34
LSTM	0.0566	36.61	4.83

Our evaluation reveals that the LSTM model demonstrated predictive accuracy compared to the FNN model, achieving a 1.51% lower relative error (4.83% vs 6.34%). This performance advantage is consistent across both evaluation metrics, with LSTM showing better results in normalized MAE (0.0566 vs. 0.0744) and original scale MAE (36.61 units vs. 48.11 units) (see Table 5.2). These findings establish LSTM as the preferred choice for this task, particularly when handling temporal or sequential data patterns. The LSTM's architectural advantage lies in its specialized gate mechanisms that effectively capture long-term dependencies and manage information flow across time sequences.

While the simpler FNN architecture offers computational efficiency, its performance limitations in handling complex temporal patterns became evident in our tests. We therefore recommend LSTM implementation for applications where temporal accuracy is critical, as the improved prediction quality justifies the additional computational requirements. However, FNN remains a viable alternative for scenarios with limited computational resources or when dealing with relatively simple temporal patterns where its faster training time might be prioritized over marginal accuracy gains. This comparative analysis suggests that model selection should be guided by both the temporal complexity of the dataset and the operational requirements for prediction accuracy versus computational efficiency.

5.7.3. Consequences for planning and network management

It is discernible that grid operators and energy planners can benefit from the forecasted load data. He also mentions that accurate load estimations enable the implementation of the predictive steps to have a reliable network [54]. It should be pointed out that big data analysis helps to enhance the efficiency of grid management and load forecasting. These estimates can help operators maximize electricity generation, minimize operating costs, employ demand response plans, and manage distribution stations on normal and in emergencies. Furthermore, one gets more elaborate monthly forecasts for long-term planning. Proper load trend forecast leads to appropriate infrastructure and maintenance plan investment decisions. This is significant for utility and grid operators who want to effectively manage the supply and demand sides.

6. CONCLUSION AND FUTURE WORK

6.1. Conclusion

Perhaps one of the most significant findings of this study and a useful contribution towards the knowledge of the correlation between electricity demand and climatic characteristics in Babylon Governorate and the strong linkage between climatic factors, which underlines the significance of incorporating weather variables into the electrical load forecasting models for better precision.

The findings suggest that the performance of models that are based on neural networks can come close to approximating the actual load – the predicted load demand map conforms well with the load demand for different months for the years 2024 and 2025, which has been deemed accurate in the period of high demand in August and January. Service providers can use these results to predict high-traffic periods and enhance how they distribute their resources, especially during the summer and winter.

6.1.1. Comparative analysis of neural network architectures

This study implemented and compared two artificial neural network models, a Feedforward Neural Network (FNN) and a Long Short-Term Memory (LSTM) network, for data analysis and prediction. The experimental results demonstrate that LSTM outperformed FNN in handling temporal patterns, achieving superior accuracy with lower error rates (4.83% vs 6.34%). The LSTM's gate mechanisms proved particularly effective for capturing sequential dependencies in the data. While FNN showed acceptable performance for simpler patterns, the findings establish LSTM as the preferred approach for time-series prediction tasks in our application domain. Future work could explore hybrid architectures to potentially combine the computational efficiency of FNN with LSTM's temporal modeling strengths.

6.1.2. Achievements

In as much as the research aimed at establishing the ability of deep neural networks to predict the load of electricity, the study achieved this goal by coming up with a strong model that could be used to predict the load of electricity. The model fared better than other approaches to analyzing data based on accuracy and its capability to work on unseen data.

6.1.3. Research limitations

That is, it was only possible to consider temperature results as the only external variable owing to the control strategies that were carried out. The result could have been even better if more parameters, such as humidity or energy costs, were incorporated.

6.1.4. Problems encountered and solutions adopted

Data accessibility and quality are two critical issues that affected the study, but they were dealt with by data cleaning and preprocessing. Moreover, changes in the topology of the neural network made it possible to minimize overfitting-related problems.

However, one can improve the model's efficiency in learning more effectively to adapt to such changes on the spot.

6.2. Future Work

It also means that to achieve reliable and comprehensive results, it is necessary to gather more profoundly significant data connected to the electrical load and climate data, for example, the level of humidity and wind direction, the rates of population growth, the data of the economic and industrial activities that might be useful for power saving in greater extent. Hybrid models can be considered as the use of large hybrid models in between educational and statistical or traditional analyses for making a fast, effective, improved long-term forecast. Examine the uses of the model in predicting winter electricity consumption and the analyzed techniques in these settings. Utility of the model with energy management systems It is suggested that the advanced model should be implemented in Energy management systems in local electrical networks, enhancing load management and cost-cutting measures.

REFERENCES

[1] D. Svozil, V. Kvasnieka, and J. Pospichal, “Introduction to multi-layer feed-forward neural networks,” vol. 39, pp. 43–62, 1997.

[2] S. Haykin, J. Nie, and B. Currie, “Neural network-based receiver for wireless communications,” 1999. doi: 10.1049/el:19990177.

[3] M. Abadi, D. Callaghan, J. Burger, and A. K. Mishra, “A machine learning approach to radar sea clutter suppression,” 2017 IEEE Radar Conf. RadarConf 2017, pp. 1222–1227, 2017, doi: 10.1109/RADAR.2017.7944391.

[4] L. M. Q. De Santana, R. M. Santos, L. N. Matos, and H. T. Macedo, “Deep Neural Networks for Acoustic Modeling in the Presence of Noise,” IEEE Lat. Am. Trans., vol. 16, no. 3, pp. 918–925, 2018, doi: 10.1109/TLA.2018.8358674.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. & Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 12, pp. 2278–2324, 1998.

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

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv. Neural Inf. Process. Syst., vol. 25, 2012.

[8] et al. Lipton, Z. C., “A critical review of recurrent neural networks for sequence learning,” 2015.

[9] S. Chen and S. A. Billings, “Neural networks for nonlinear dynamic system modeling and identification,” Int. J. Control, vol. 56, no. 2, pp. 319–346, 1992, doi: 10.1080/00207179208934317.

[10] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks: The state of the art,” Int. J. Forecast., vol. 14, no. 1, pp. 35–62, 1998, doi: 10.1016/S0169-2070(97)00044-7.

[11] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, 2013.

[12] N. Amjady, F. Keynia, and H. Zareipour, “Short-term load forecast of microgrids by a new bilevel prediction strategy,” IEEE Trans. Smart Grid, vol. 1, no. 3, pp. 286–294, 2010, doi: 10.1109/TSG.2010.2078842.

[13] T. Hong and S. Fan, “Probabilistic electric load forecasting: A tutorial review,” Int. J. Forecast., vol. 32, no. 3, pp. 914–938, 2016, doi: 10.1016/j.ijforecast.2015.11.011.

[14] M. S. Kandil, S. M. El-Debeiky, and N. E. Hasanien, “Long-term load forecasting for fast developing utility using a knowledge-based expert system,” IEEE Trans. Power Syst., vol. 17, no. 2, pp. 491–496, 2002.

[15] J. W. Taylor and P. E. McSharry, “Short-term load forecasting methods: An evaluation based on European data,” . IEEE Trans. Power Syst., vol. 22, no. 4, pp. 2213–2219, 2007.

[16] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks: The state of the art,” Int. J. Forecast., vol. 14, no. 1, pp. 35–62, 2013.

[17] Y. Wang, M. Liu, Z. Bao, and S. Zhang, “Short-term load forecasting with multi-source data using gated recurrent unit neural networks,” Energies, vol. 11, no. 5, p. 1138, 2018.

[18] Y. Xiao, Y. Xu, J. Zhou, J. Chen, and Y. & Wang, “A hybrid deep learning-based model for short-term load forecasting using empirical mode decomposition and convolutional neural network,” Energy, vol. 202, p. 117708, 2020.

[19] S. Yadav, M. Tomar, and A. K. Yadav, “Hybrid model using LSTM and CNN for short-term load forecasting,” J. Energy Storage, vol. 41, p. 102862, 2022.

[20] K. Chen, J. Chen, and X. He, “Short-term electrical load forecasting using the support vector regression (SVR) model to calculate the price of electricity in the south of Brazil,” Energy, vol. 118, pp. 774–785, 2017.

[21] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network,” 2017.

[22] C. Wen, K. Zhou, S. Yang, and L. & Li, “A review of big data analytics for electric load forecasting,” IEEE Trans. Power Syst., vol. 34, no. 4, pp. 3413–3424, 2019.

[23] N. Amjady, “Short-term hourly load forecasting using time-series modeling with peak load estimation capability,” IEEE Trans. Power Syst., vol. 16, no. 4, pp. 798–805, 2001.

[24] S. Fan and R. J. Hyndman, “Short-term load forecasting based on a semi-parametric additive model,” IEEE Trans. Power Syst., vol. 27, no. 1, pp. 134–141, 2012.

[25] 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.

[26] S.-H. Chen, “Forecasting electricity demand using neural networks,” Energy Convers. Manag., vol. 45, no. 3, pp. 287–296, 2004.

[27] M. Marvasti, Z. Tan, R. Ghaemi, and A. M. & Foley, “Hourly electricity demand forecasting in buildings using neural networks,” Energy Build., vol. 138, pp. 287–294, 2017.

[28] F. Yu, J. Deng, and H. & Wang, “A hybrid deep learning model for short-term electric load forecasting,” Energy, vol. 142, pp. 48–56, 2017.

[29] W. Kong, Z. Y. Dong, and Y. & Jia, “Short-term residential load forecasting based on LSTM recurrent neural network,” IEEE Trans. Smart Grid, vol. 10, no. 1, pp. 841–851, 2019.

[30] Y. Wang, N. Zhang, and Y. & Ren, “Transfer learning for short-term wind power forecasting with deep neural networks,” Renew. Energy, vol. 149, pp. 246–256, 2020.

[31] T. Hong, P. Pinson, and S. & Fan, “Global energy forecasting competition 2012: Results and future perspectives,” Energy, vol. 93, pp. 172–195, 2016.

[32] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[33] J. Park and I. W. Sandberg, “Universal Approximation Using Radial-Basis-Function Networks,” Neural Comput., vol. 3, no. 2, pp. 246–257, 1991, doi: 10.1162/neco.1991.3.2.246.

[34] D. Li and R. & Cao, “Long-term load forecasting using support vector machines with hybrid optimization algorithms.,” Appl. Soft Comput., vol. 29, pp. 287–292, 2015.

[35] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: A new learning scheme of feedforward neural networks,” IEEE Int. Conf. Neural Networks - Conf. Proc., vol. 2, pp. 985–990, 2004, doi: 10.1109/IJCNN.2004.1380068.

[36] L. Yu, K. Y. Lee, and and S. H. Kim, “Load forecasting with neural networks,” IEEE Trans. Power Syst., vol. 7, no. 1, pp. 151–157, 1992.

[37] A. Marino, M. Stojanovic, and and A. A. Palazoglu, “Load forecasting for energy management using a hybrid neural network and statistical approach,” Energy Reports, vol. 2, pp. 159–166, 2016.

[38] S. Gensler, F. H. A. A. Vieira, and and M. P. F. Silva, “Load forecasting in smart grids using neural networks: A review and new approaches,” Renew. Sustain. Energy Rev., vol. 62, pp. 425–438, 2016.

[39] W. He, “Load Forecasting via Deep Neural Networks,” Procedia Comput. Sci., vol. 122, pp. 308–314, 2017, doi: 10.1016/j.procs.2017.11.374.

[40] I. Guyon and A. & Elisseeff, “An introduction to variable and feature selection,” J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.

[41] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” Comput. Electr. Eng., vol. 40, no. 1, pp. 16–28, 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[42] K. Kira and L. A. & Rendell, “A practical approach to feature selection,” Proc. Ninth Int. Conf. Mach. Learn., pp. 249–256, 1992.

[43] M. Kuhn and K. Johnson, Applied predictive modeling. 2013. doi: 10.1007/978-1-4614-6849-3.

[44] C. M. Bishop, “Pattern Recognition and Machine Learning,” Springer, 2006.

[45] T. Hastie, “The elements of statistical learning: data mining, inference, and prediction,” 2009, Springer.

[46] J. Han, M. Kamber, and J. & Pei, "Data Mining: Concepts and Techniques," Morgan Kaufmann, 2011.

[47] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[48] R. J. A. Little and D. B. & Rubin, "Statistical Analysis with Missing Data," Wiley, 2019.

[49] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," J. Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[50] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," O'Reilly Media, 2017.

[51] S. Makridakis, S. C. Wheelwright, and R. J. & Hyndman, "Forecasting: Methods and Applications," Wiley, 1998.

[52] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," Int. J. Forecast., vol. 22, no. 4, pp. 679–688, 2006, doi: 10.1016/j.ijforecast.2006.03.001.

[53] N. R. Draper and H. & Smith, "Applied Regression Analysis," Wiley, 1998.

[54] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A Multi-Horizon Quantile Recurrent Forecaster," no. Nips 2017, 2017, [Online]. Available: http://arxiv.org/abs/1711.11053

[55] American Public Power Association, "Substation 33/11k.v for electricity distribution, " 2017, Unsplash , Available: https://unsplash.com/photos/VuR4oHZ3ucc

[56] Nikhil Shahu, "33kV&11kV Circuit breakers," 2019, Unsplash, Available at: https://unsplash.com/photos/gray-machine-s89Z5V0plTY.

[57] Frantisek Duris, "33kv&11kv Circuit breakers," 2025, Unsplash,Available at: https://unsplash.com/photos/a-large-control-room-with-lots-of-control-knobs-C3DfIgig1j8.

CURRICULUM VITAE

Name Surname : Mudhar ABDULHADI

EDUCATION:

- **Graduate** : 2024, Sakarya University, Computer and Information Engineer, Computer Engineering
- **Undergraduate** : 2017, Al-Mustaqbal University, Computer and Network Technology Engineering
- **Diploma** : 2004, Institute of Technology - Baghdad, Surveying Department

PROFESSIONAL EXPERIENCE AND AWARDS:

- Technical Supervisor in Babylon Electricity Distribution - In the field of project implementation and maintenance since 2006
- I worked in the General Directorate of Middle Euphrates Electricity Distribution - In the field of planning and studies
- I worked as an engineer in the field of research and development - Head of the Statistics Department since 2017
- I received many letters of thanks and gratitude from the University of the Future as I was ranked second in the Department of Computer Engineering and for the second and third stages in 2015-2016
- I received many letters of thanks and appreciation and financial and appreciative rewards from the Ministry of Electricity and the General Company for the Distribution of Middle Euphrates Electricity.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Abdulhadi M., Zengin A. (August 25-26, 2024). Development of electrical load forecasting using neural networks. The 5th International Conference on Engineering and Applied Natural Sciences, Konya, Turkey.