

# Department of Electrical & Electronics Engineering

PROJECT REPORT

on

## Load Forecasting in Smart Grid using Deep Learning Methods

*Submitted in partial fulfillment of the requirements for the award of the degree of*

**Bachelor of Engineering in**

**Electrical & Electronics Engineering**

*Submitted By*

## SUMAN MONDAL

**1NT19EE099**

Project Carried out at

***Department of Electrical & Electronics Engineering, NMIT***

*Under the guidance of*

## Internal Guide: External Guide:

**Ms. Shruti Gatade**

**Assistant Professor,**

**Dept. Of Electrical & Electronics Engineering,**

**NMIT, Bengaluru**

**2022-2023**



Department of Electrical and Electronics Engineering

# CERTIFICATE

This is to certify that the Project report entitled **“**Load Forecasting in Smart Grid using Deep Learning Method**”**is carried out at Nitte Meenakshi Institute Of Technology **,** by **Suman Mondal** bearing **USN:1NT19EE099** a Bonafide student of Nitte Meenakshi Institute of Technology in partial fulfillment for award of degree in Bachelor of Engineering in **Electrical & Electronics Engineering** of the Visvesvaraya Technological University during the academic year 2022-2023. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report and deposited in the departmental library. The report has been approved as it satisfies the academic requirements for completion of the autonomous scheme of Nitte Meenakshi Institute of Technology for the above-said degree.

| **Name & Signature of Name & Signature of**  **Internal guide Head of the Department** | |
| --- | --- |
| **Principal** | |
| **External Viva** | |
| **Name of the Examiners** | **Signature with Date** |
| **1.** |  |
| **2.** |  |



Department of Electrical and Electronics Engineering

# DECLARATION

I hereby declare that the Project report entitled **“**Load Forecasting in Smart Grid using Deep Learning Methods**”** has been carried out by me and submitted in partial fulfillment of the course requirements for the award of degree in **Bachelor of Engineering** in Electrical & Electronics Engineering during the academic year **2021-2022**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree or diploma.

**Place: Bengaluru Name of Student:**

**Suman Mondal**

**Date: USN:**

**1NT19EE099**



# Acknowledgement

The successful execution of our project gives us an opportunity to convey our gratitude to each one who has been instrumental in paving the path to our continuation of this project. Whatever we have done is due to such guidance and help and we would not forget to thank them all.

We wholeheartedly thank our guide **Mrs. Shruti Gatade,** Assistant Professor, Department of Electrical and Electronics Engineering, for her support and guidance anytime we required.

We express our deep sense of gratitude to **Dr. Premila Manohar**, HoD, Department of Electrical and Electronics Engineering, for her kind cooperation, valuable guidance and creating the best learning environment for us.

We would like to express our gratitude to the **Nitte Meenakshi Institute of Technology** and our beloved Principal **Dr. H C. Nagaraj** for providing us the support, facilities and motivation to carry out our project.

We also thank and share this moment of happiness with our parents who rendered us enormous support during the whole tenure of our studies at Nitte Meenakshi Institute of Technology, Bengaluru.

Finally, we would like to thank all the others who helped us in various ways to gain knowledge and have good training.

|  |  | | | Aman Raj  M S Sreekar  Rishab K Mogral  Suman Mondal | | (1NT19EE012) (1NT19EE055) (1NT19EE082) (1NT19EE099) |
| --- | --- | --- | --- | --- | --- | --- |
| Place: Bengaluru Date: | |  |  | |

**Abstract**

A crucial step in smart networks is load forecasting, which includes estimating future power consumption. Load forecasting helps utilities to foresee and prepare for the power requirements of a certain region or set of consumers by analyzing historical data, weather patterns, customer information, and socio-economic factors. While medium- and long-term predictions aid in the optimisation of power generation, distribution, and maintenance activities, short-term forecasts assist operators in maintaining the grid's balance in real-time. In the dynamic environment of the modern power grid, accurate load forecasting provides effective resource allocation, cost savings, and system stability. A considerable transition from outdated approaches to new ones has occurred in the application of deep learning algorithms in load forecasting. Traditional strategies depended on presumptions about data distribution and linear correlations, including statistical techniques like regression models and time series analysis. These techniques frequently have trouble capturing the intricate patterns and non-linearities found in load data. Older methods also needed manual feature engineering, which was laborious and subject to human biases. Recurrent neural networks (RNNs) containing Long Short-Term Memory (LSTM) cells, on the other hand, have become effective tools for load forecasting using contemporary deep learning approaches. RNNs are efficient in capturing temporal relationships and long-term trends, making forecasts more precise and adaptable.The goal of this research is to add to the corpus of knowledge on deep learning-based load forecasting in smart grids. The suggested models and the combined strategy provide promising answers for precisely forecasting load demand. The results shed light on the efficacy of LSTM, DenseNet, and ResNet models in load forecasting as well as the advantages of combining different methods, such as SVM and XG-DTC, for improved prediction accuracy.The project's findings can help academics and grid operators choose the best deep learning models for load forecasting jobs in smart grid systems. Smart grids can optimize their operations, increase energy efficiency, and guarantee dependable and stable power distribution by properly forecasting load demand.



|  | | **Contents** |  |
| --- | --- | --- | --- |
| **Acknowledgement** |  |  | . |
| **Abstract** |  |  |  |
| [**List of Figures**](#_1fob9te) |  |  |  |
| [**List of Table**](#_2et92p0) |  |  |  |
| **Chapter 1**  1.1 Motivation | **Introduction** |  |  |

1.2 Organization of the Report

**Chapter 2 Literature Survey**

* 1. Background Work
  2. Open Issues and Challenges
  3. Problem Definition
  4. Objectives
  5. Scope of the Work

**Chapter 3 Methodology**

3.1 Model 1 : LSTM RNN with GRNN

3.2 Model 2 : DenseNet and ResNet

3.3 Model 3: SVM with XG-DTC

**Chapter 4 Implementation Details**

4.1 LSTM RNN Model

4.2 DenseNet and ResNet Model

4.3 SVM with XG-DTC model

4.4 Software Used.

**Chapter 5 Result and Analysis**

**Chapter 6 Conclusion and Future Scope**

**Bibliography**



# List of Figures

**Fig. 1.** The structure of an LSTM block

**Fig. 2**. The overall architecture of the densely connected Network

**Fig.3.** Daily and Weekly load demand analysis

**Fig.4**. Monthly Load Analysis

**Fig.5.** Plot Predictions for Original, Training and Test Data

**Fig. 6.** Plot Predictions for training, test and forecast data

**Fig. 7.** Predicted Data requests for the next 720 hours

**Fig. 8.** Predicted Data requests for the day on hourly basis

**Fig. 9.** GRNN predictions

**Fig. 10.** Predicted load using DenseNet and ResNet

**Fig. 11.** SVM Load forecasting - Actual vs. Predicted Data

**Fig. 12.** XG-Boost Load forecasting - Actual vs. Predicted Data

**Fig. 13.** Decision Tree Load forecasting - Actual vs. Predicted Data

**Fig. 14.** RMSE,MAE and Accuracy for RNN and LSTM

**Fig.15.** RMSE AND MAE Values for DenseNet and ResNet

**Fig.16**. RMSE AND MAE Accuracy Values for DenseNet and ResNet

**Fig.17.** RMSE AND MAE Values for SVM,DTC and XgBoost

**List of Tables**

TABLE I :RMSE, MAE OF MODEL

TABLE II: ACCURACY OF BASE, OVERSAMPLED & UNDERSAMPLED DATASET

TABLE III: RMSE & MAE OF THE MODEL

TABLE IV: RMSE, MAE AND ACCURACY OF MODEL

TABLE V: RMSE, MAE AND ACCURACY OF MODEL

TABLE VI: ACCURACY OF BASE, OVERSAMPLED & UNDERSAMPLED DATASET

TABLE VII: RMSE & MAE OF THE MODEL

**Chapter 1**

**Introduction**

**1.1 Motivation**

The development of smart grid technology could be a response to the growing need for efficient and sustainable energy management. By integrating cutting-edge communication, sensor, and control technologies into traditional power networks, smart grids enable improved monitoring, control, and optimisation of energy generation, transmission, and consumption. To fully utilize the potential of smart grids, accurate load forecasting is necessary for successful power system planning, energy scheduling, and resource allocation.

The significance of load forecasting in the context of smart grids is what spurred the development of this project. Utilizing cutting-edge technology and data-driven strategies, smart grids seek to optimize the production, distribution, and use of electricity. In order for utilities to make educated decisions about energy generation, load balancing, and demand response techniques, accurate load forecasting is a crucial part of smart grid operations.

Traditional load forecasting techniques frequently rely on statistical models that could find it difficult to accurately represent the complexity and non-linear patterns found in huge datasets. On the other hand, deep learning approaches have shown considerable promise in many domains due to their capacity to understand complex relationships within data and to develop hierarchical representations. As a result, there is a chance to increase forecasting precision and improve grid management by using deep learning techniques into smart grid load forecasting.

Making forecasts about future energy consumption based on past load data and other essential variables like the weather, holidays, and economic indicators is a crucial task in power system management known as load forecasting. With the use of precise demand projections, power system operators can optimize energy production and distribution, reduce costs, and ensure grid stability.

Utilities can gain from more precise predictions by creating a deep learning-based load forecasting system, which will increase operational effectiveness, save costs, and better use energy resources. The goal of this project is to investigate and present the potential of deep learning algorithms for load forecasting tasks in the context of smart grids. It aims to overcome the drawbacks of conventional forecasting techniques and offer information on the use of deep learning models for load forecasting in practical settings.

Machine learning techniques have gained a lot of attention in the field of load forecasting due to their ability to discern complex patterns and correlations in large datasets.

We present a comparative analysis of load forecasting models using three deep learning architectures: Recurrent Neural Network-Long Short-Term Memory (RNN-LSTM), DenseNet, and ResNet, as well as a hybrid model that combines Support Vector Machines (SVM), eXtreme Gradient Boosting (XGBoost), and Decision Tree Classifier (DTC).

These models are evaluated based on their accuracy in projecting future load demand and whether or not they are suitable for usage with the smart grid.

The RNN-LSTM model performs exceptionally well in load forecasting jobs because it can identify temporal correlations and long-term patterns in time series data. By using its recurrent nature and memory cells, RNN-LSTM can effectively reflect the dynamic nature of load consumption by taking into consideration historical load values and their sequence.

In contrast, feature extraction tasks are best handled by the deep convolutional neural network architectures DenseNet and ResNet. These algorithms may uncover intricate relationships between load profiles and auxiliary information, including weather, which improves forecasting accuracy.

To explore other tactics, we propose a hybrid model that incorporates the advantages of SVM, XGBoost, and DTC. By combining the nonlinear relationship handling capabilities of SVM, the robust gradient boosting approach of XGBoost, and the decision-making skills of DTC, the ensemble model's goal is to increase the accuracy and robustness of load forecasting.

These models are assessed and compared using a real-world load dataset, accounting for performance metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).Computational effectiveness and model complexity are also taken into account when determining if a model is practical for usage in smart grid systems. Despite coming at the performance of the model from slightly different angles, both metrics offer insightful information. The RMSE calculates the average discrepancy between the expected values and the actual values. Calculated is the squared mean discrepancy between the expected and actual values. The RMSE provides a measurement of the extent of errors, with lower values suggesting greater accuracy. Considering both positive and negative errors, the RMSE provides a complete picture of the entire predicting performance.

The perspectives offered by RMSE and MAE on a load forecasting model's accuracy are complementary. While RMSE provides information on the amount of errors, MAE provides an average measure of correctness. By considering both metrics, researchers and practitioners may fully understand the model's performance and decide whether it is suitable for a certain application.

The findings of this study provide important light on the advantages and disadvantages of various load forecasting models, particularly in the context of smart grids. Power system operators may efficiently manage energy generation, transmission, and consumption by using precise load forecasting techniques, which improve grid efficiency, lower costs, and increase sustainability.

Additionally, this project aims to advance smart grid technology research and development. It can progress the field and promote future investigation of novel methodologies by examining the difficulties and potential connected with deep learning-based load forecasting. The ultimate goal of this project is to accelerate the transition to smart grids that are more effective, sustainable, and dependable by applying cutting-edge deep learning approaches to load forecasting.

**1.2 Organization of report**

Following is an organization of this report on deep learning-based load forecasting for smart grids. Background information is given in the introduction, along with highlights of the project's inspiration and goals. A summary of load forecasting methods within the framework of smart grids is provided in the literature review section, which also explores the use of deep learning techniques. Additionally, it covers earlier research that employed Decision Tree Classifier, LSTM, DenseNet, ResNet, SVM, and XGBoost for load forecasting.

The dataset utilized in the work, which includes load data and pertinent meteorological data, is described in the data collecting and preparation section. It describes the stages involved in data collecting and preparation, including feature engineering, normalization, and data cleaning. The LSTM, DenseNet, and ResNet models, as well as the combined model that uses SVM, XGBoost, and Decision Tree Classifier, are all thoroughly explained in the methods section. It also describes the training process for each model and the performance evaluation measures.

The experimental setup and the findings from each deep learning model are covered in the experimental results portion of the study. It examines performance measures like mean absolute error (MAE) and root mean square error (RMSE), analyzes the findings, and then offers conclusions and observations. The performance of the various models is compared in the discussion part that follows, along with their advantages and disadvantages, and the viability and efficiency of deep learning techniques for load forecasting in smart grids is assessed.

The aims, techniques, and major conclusions of the project are summarized in the conclusion section. It examines the study's consequences and importance and makes suggestions for further study and improvements to the load forecasting algorithms. A references section that lists all of the mentioned sources used in the project report appears at the end of the report.

**Chapter 2**

**Literature Survey**

**2.1 Background Work**

In order for smart grids to operate and be managed effectively, load forecasting is essential. Many methods have been used over the years to precisely anticipate power consumption, from conventional statistical models to cutting-edge machine learning algorithms. Due to its capacity to recognise intricate connections and patterns in time series data, deep learning techniques have recently attracted a lot of attention in the field of load forecasting.

The Long Short-Term Memory (LSTM) model is one of the extensively used deep learning architectures for load forecasting. In terms of processing sequence data and capturing long-term dependencies, LSTM has produced encouraging results. It can record seasonal and temporal trends in load data because it uses memory cells to store information for long periods of time. Numerous studies have shown that LSTM performs better than previous approaches in load forecasting jobs by obtaining reduced prediction errors.

Other deep learning architectures have also been investigated for load forecasting in smart grids in addition to LSTM. In order to capture geographical correlations and extract pertinent information from load and weather data, DenseNet, a densely linked convolutional neural network, has demonstrated potential. Due to the effective parameter sharing and feature reuse made possible by its distinctive design, forecast accuracy has increased. Similar to ResNet, a deep residual network, residual connections have shown to be useful in learning complicated relationships. ResNet has been effectively used in load forecasting, producing results that are competitive and revealing information about temporal dynamics.

Despite the potential of deep learning models, integrating several methods can improve the accuracy of load predictions even further. One strategy combines deep learning models with conventional machine learning methods. Such algorithms that have been effectively integrated with deep learning techniques include Support Vector Machine (SVM), XGBoost, and Decision Tree Classifier. This combination strategy makes use of the advantages of many models to identify both global and local trends, enhancing the performance of prediction as a whole.

The usefulness of deep learning techniques for load forecasting in smart grids has been confirmed by prior investigations. The prediction accuracy, computational effectiveness, and scalability of many models, including LSTM, DenseNet, ResNet, SVM, XGBoost, and Decision Tree Classifier, have been compared by researchers. To further improve forecasting performance, they have investigated a variety of preprocessing methods, feature engineering methodologies, and model optimisation procedures.

[1] In order to demonstrate the value gained, they present original data pre-processing methods in this study. They then apply these methods to the training data for an ANN and assess the precision of the predictions it produces.Forecasting load is essential for supply and demand balancing and for setting power prices. It has been common practice to utilize artificial neural networks (ANNs), which can forecast future loads after being trained on a set of data. The training set of data's quality and availability has an impact on how successfully the ANNs predict outcomes. The suggested strategies and the results are shown using usage data from the Greek interconnected power grid.

[2] In this paper they Used an extreme learning machine (ELM) with autonomous knowledge representation from a particular input-output data set, this research explores short-term electricity-load forecasting. While the traditional ELM runs without knowledge information, they employed a Takagi-Sugeno-Kang (TSK)-based ELM to provide a methodical technique to producing if-then rules. For predicting, the tests were run using short-term electrical load data. With the use of historical loads from the New England ISO, weather predictions, holiday information, and electricity-load data, hourly day-ahead loads were predicted. They employed measurements and statistical traits including root mean squared error (RMSE), mean absolute error (MAE), mean absolute percent error (MAPE), and R-squared, respectively, to measure the forecaster's performance. The experimental findings demonstrated that the suggested.When compared to a traditional ELM with four activation functions—sigmoid, sine, radial basis function, and rectified linear unit (ReLU)—the technique performed well. It had a limited number of rules, excellent knowledge information, and prediction performance.

[3] In this paper they Used an extreme learning machine (ELM) with autonomous knowledge representation from a particular input-output data set, this research explores short-term electricity-load forecasting. While the traditional ELM runs without knowledge information, they employed a Takagi-Sugeno-Kang (TSK)-based ELM to provide a methodical technique to producing if-then rules. For predicting, the tests were run using short-term electrical load data. With the use of historical loads from the New England ISO, weather predictions, holiday information, and electricity-load data, hourly day-ahead loads were predicted. They employed measurements and statistical traits including root mean squared error (RMSE), mean absolute error (MAE), mean absolute percent error (MAPE), and R-squared, respectively, to measure the forecaster's performance. The experimental findings demonstrated that the suggested.When compared to a traditional ELM with four activation functions—sigmoid, sine, radial basis function, and rectified linear unit (ReLU)—the technique performed well. It had a limited number of rules, excellent knowledge information, and prediction performance.

[4]Developed a multimodal Recurrent Neural Network (m-RNN) model for creating creative picture captions in this study. The probability distribution of creating a word given prior words and a picture is explicitly modeled. Captions for images are created by randomly selecting from this distribution. A deep recurrent neural network for texts and a deep convolutional network for pictures make up the model. The entire m-RNN model is made up of these two sub-networks and their multimodal layer of interaction. Four benchmark datasets are used to validate the performance of our model. Modern techniques are outperformed by our model. Additionally, we use the m-RNN model for retrieval tasks to retrieve words or images, and it significantly outperforms state-of-the-art techniques that just focus on ranking retrieval functions that are objective.

[5] The voice recognition system described in this research immediately transcribes audio input into text without the need for a phonetic representation in between. The deep bidirectional LSTM recurrent neural network architecture and the Connectionist Temporal Classification goal function provide the system's foundation. The goal function is modified, and the network is trained to minimize the expectation of any transcription loss function. This enables the word mistake rate to be directly optimized, even in the absence of a lexicon or language model.

[6] The demand for electricity has rapidly increased as a result of economic and social development. The change of the household energy consumption structure and the reduction of global warming are aided by accurate residential power load forecasting. In order to improve prediction accuracy, the dilated convolutional neural network (DCNN), long short-term memory network (LSTM), autoencoder (AE), and attention mechanism (AM) are combined in the hybrid residential short-term load forecasting framework (DCNN-LSTM-AE-AM) proposed in this paper. First, the original data is preprocessed using a T-nearest neighbors (TNN) technique. A DCNN is further added to extract the long-term characteristic. To learn the sequence features concealed in the extracted features and decode them into output features, we secondly combine the LSTM with the AE (LSTM-AE). At last, the AM is also added in order to extract and combine the high-level stage information to provide the prediction outcomes. The suggested technique outperforms previous methods and is effective at capturing the oscillation features of low-load data, according to experiments on two real-world datasets.

[7] This article outlines a technique for enhancing peak load forecasting and a day-ahead 24-hour load curve's performance. Using the optimal design parameters, radial basis function (RBF) neural network models are used to predict the next-day load curve. The weighted sum of the error of the current prediction and the change in errors between the current and prior prediction is applied to the load curve projected using RBF network models in order to increase forecasting accuracy. Using differential evolution, the best coefficients for lowering the mean absolute percent error (MAPE) and the total error are also found. Using four years' worth of hourly load data received from the Korea Power Exchange, the suggested models are trained and evaluated.

[8] The most recent and well-liked deep learning approach, long short-term memory (LSTM) recurrent neural network (RNN), is what we suggest in this research as a framework to address this challenging problem. The suggested framework is put to the test using a collection of actual household smart meter data that is publicly available, and the results are thoroughly compared to several benchmarks, such as the state-of-the-art in load forecasting. As a consequence, when it comes to short-term load forecasting for specific residential families, the suggested LSTM technique performs better than the other stated competition algorithms.

[9] The short-term load forecasting problem is addressed in this study using a method based on ensemble learning.The acquired experimental findings demonstrate that a short-term power consumption forecasting strategy based on ensemble learning can assist in merging predictions made by less effective learning methods in order to generate better outcomes. In particular, compared to other state-of-the-art methods applied to the same dataset, the system yields a reduced error. More significantly, this case study has demonstrated that employing an ensemble technique may produce extremely precise forecasts, demonstrating that it is an appropriate strategy for tackling the short-term load forecasting problem.

[10] This work discusses the ELM's power engineering applications before creating an ELM-based predictor for power systems' real-time frequency stability assessment (FSA). Power system operating parameters are the predictor's inputs, and its output is the frequency stability margin. ELM's quick speed enables the predictor to be updated live for improved robustness and dependability.

[11] A two-stage short-term load forecasting (STLF) model for power transformers is suggested in this research. 1) The aggregated substation-level demand is predicted using three cutting-edge methods, with historical load, weather, and calendar data serving as inputs. Since no unique STLF model needs to be created at this time, forecasters can choose the most accurate prediction outcomes for predicting transformer-level load. 2) The ratio of the transformer load to the substation load is known as the load distribution factor (LDF). Under various substation operating circumstances, nonlinear regression functions reflect the link between LDF and substation load.

[12] In order to improve the robustness and applicability of the algorithm, we present a technique for forecasting the power consumption in the UK using multivariate k-NN Regression, coding working days against non-working and weekend days as binary dummy variables. Both the suggested univariate and multivariate k-NN extensions are evaluated.

[13] Forecasting electricity demand has proven to be a significant difficulty for power system scheduling across a range of energy industries. The electrical market has used a variety of artificial intelligence approaches and procedures for short-term load forecasting, but there is little information regarding their viability given the type of data and other relevant variables. This paper offers numerous scientific and technological justifications for short-term load forecasting techniques based on the findings of earlier energy researchers. To illustrate how effective each strategy is in different situations, the fundamental advantages and disadvantages of these approaches are presented. A hybrid method is then suggested.

[14] This research suggests a novel big data-based short-term load forecasting approach. Using data from smart meters, a cluster analysis is first carried out to categorize daily load patterns for specific loads. The next step is to identify the key influential elements using an association analysis. A decision tree is then used to build categorization rules after that. The best forecasting models are then selected for the various load patterns. Using actual load data, it is demonstrated that the new framework under consideration can provide short-term load forecasting accuracy within predetermined bounds.

[15] In contrast to conventional aggregated system-level load forecasting, the AMI data offers a new perspective on how load forecasting is carried out, from very short-term load forecasting to long-term load forecasting at system, regional, feeder, or even consumer level. The work done to identify groups of customers that have similar load consumption patterns using smart meters in order to improve system level intraday load forecasting before load forecasting

[16] Here, approaches for forecasting demand from the bottom up are suggested, and they make use of both improved classification/clustering models and existing forecasting techniques. The Practise Theory of human behavior is used to develop a Markov Chain based sampling approach, which is suggested as a way to provide a prediction with little computing work and less need for past data. Seasonal adjustments and environmental variables are not necessary for the suggested modeling methodology. The forecast and actual demand for a cohort of residential loads over a 5-month period are used to assess various models and show that using an ensemble forecast may significantly enhance performance.

[17] We are interested in predicting household-level power consumption since it is crucial to maintaining supply and demand balance in the LV network. A brand-new approach to forecasting short-term functional time series has been developed. The effectiveness of the suggested approach to estimate the intra-day home level load curves was demonstrated by an application to the Irish smart meter data set.

[18] This study investigates the possible effects of automated meter reading (AMR) on a residential customer's short-term load forecasting. A utility business models the real-time measurement data it receives from customers' smart meters as the combination of a deterministic component and a Gaussian noise signal. Utilizing spectral analysis, the shaping filter for Gaussian noise is determined. Then, load prediction is performed using Kalman filtering. The suggested method's accuracy is assessed for various sample intervals and planning horizons. The findings indicate that the accuracy of the load forecast is much enhanced with the availability of additional real-time measurement data.

[19] In this paper,In order to show the value gained, they present unique data pre-processing techniques in this research, which they then apply to the training data for an ANN and evaluate the accuracy of the predictions it generates.For the purpose of balancing supply and demand and establishing power pricing, load forecasting is crucial. Artificial neural networks (ANNs), which can anticipate future loads after being trained on a collection of data, have often been used to support load forecasting. The caliber and accessibility of the training set of data affects how well the ANNs anticipate outcomes. Utilizing consumption information from the Greek interconnected power system, the suggested tactics and the outcomes are demonstrated.

[20] Two forecasting models for short-term electrical load are created in this research utilizing long short term memory neural networks (LSTM NN). While the second model anticipates multi-step intraday rolling horizons, the first model predicts a single step forward load. In addition to the meteorological information for the area under consideration, the load time series is used. It is demonstrated that embedding a rolling time-index series that includes a day of the week index, a holiday flag, and a time of day index as a categorical feature vector considerably improves forecasting accuracy. Additionally, the performance of other machines, specifically a generalized regression neural network (GRNN) and an extreme learning machine (ELM), is also presented to assess the performance of the LSTM NN. The suggested algorithms are evaluated using hourly load data from the Electrical Reliability Council of Texas (ERCOT), which serves as benchmark data.

**2.2 Open Issues and Challenges**

There are a number of unresolved problems and difficulties that academics and practitioners need to address even if the use of deep learning approaches for load forecasting in smart grids shows promise. These difficulties include:

Data Quality and Availability: Finding sufficient, high-quality data to train and test deep learning models is one of the major issues. It might be difficult to find precise and trustworthy load data as well as pertinent weather data. Processes for gathering data should make sure to include a variety of variables that affect load demand, such as holidays, special occasions, and anomalies. Additionally, to deal with missing values, outliers, and inconsistencies, data preparation techniques must be used.

Feature Engineering: Deep learning models frequently need rigorous feature engineering to extract pertinent features from the input data. The performance of the model may be greatly improved by selecting the most useful elements and correctly modifying them. It is still difficult to develop automated feature engineering methods or integrate domain knowledge into the feature selection process.

Model Interpretability: Because deep learning models are sometimes referred to as "black-box models," it might be challenging to understand the predictions they provide. Making decisions on the management of the smart grid may need an understanding of the variables and factors that affect load forecasts. Research is still being done on methods for deciphering the predictions of deep learning models and giving insights into the underlying linkages and patterns.

Model Generalization:Deep learning algorithms can overfit, particularly when the training data is small or not reflective of real-world situations. It might be difficult to make sure that the trained models can generalize to new data and adjust to shifting load patterns. This problem may be solved with the use of methods like regularization, cross-validation, and transfer learning.

Efficiency of computation: Deep learning models can be computationally expensive and demand a lot of computer power, especially when used to solve complex load forecasting issues in smart grids. Computational constraints can be solved by creating effective methods and algorithms to reduce the lengths of model training and inference processes as well as by investigating hardware accelerations.

Scalability and Model Integration: It is difficult to scale deep learning models for use in real-time load forecasting within the framework of current smart grid technologies. Important factors to take into account include modifying the models to accommodate streaming data, resolving synchronization and communication problems, and creating effective deployment plans.

Factors of Regulation and Economy: The application of deep learning techniques for load forecasting in smart grids may present problems of regulation and economy. Important factors to think about when deploying deep learning-based load forecasting systems include ensuring regulatory compliance, resolving privacy issues, and determining the cost-effectiveness and return on investment.

**2.3 Problem Statement**

This project uses deep learning techniques to solve the issue of precise load demand forecasting in smart grids. For effective grid management, resource planning, and decision-making, load forecasting is essential. For load forecasting, conventional statistical models and machine learning algorithms have been widely used; nevertheless, deep learning techniques have drawn interest because of their capacity to recognise intricate connections and patterns in time series data.

The creation and use of deep learning models for load forecasting in smart grids, such as LSTM, DenseNet, ResNet, and a combined model of SVM, XGBoost, and Decision Tree Classifier, are among the particular issues that need to be addressed. These models must be able to correctly forecast future load demand using previous load data and pertinent meteorological data. The study intends to investigate the benefits and drawbacks of various deep learning architectures and evaluate how well they execute load forecasting tasks.

The initiative has difficulties related to data quality and availability as well. In order to train and test the models, it is crucial to have precise and trustworthy load data as well as meteorological data. To deal with missing values, outliers, and inconsistencies in the data, preprocessing procedures must be used. In order to increase the forecasting accuracy, feature engineering is also essential for removing useful elements from the incoming data.

The research must also handle the deep learning models' interpretability. For the purpose of making decisions and managing the grid, it is crucial to comprehend the elements and aspects that go into load forecasting. A crucial component of the study is developing methods for interpreting model predictions and giving insights into underlying patterns and linkages.

In general, the problem statement concentrates on utilizing deep learning techniques to precisely estimate load demand in smart grids, while addressing issues with data accessibility, feature engineering, model interpretability, and model performance assessment. By tackling these issues, the project hopes to develop load forecasting methods and make it possible for smart grid management to be more effective and efficient.

**2.4 Objective**

In the context of the smart grid, load forecasting strives to deliver precise, trustworthy, and scalable estimates of future electricity consumption. Enabling effective energy management, facilitating the integration of renewable energy sources, and ensuring grid stability are the main goals. The upcoming factors must be taken into account in order to reach these objectives:

i)To account for modifications in electricity usage, changes in the weather, and other pertinent aspects, the forecasting models must be updated in real-time. Operators can make wise decisions and successfully handle changes in demand by continuously adapting to the status of the system.

ii)In load forecasting, accuracy and dependability are essential components. The models should work to provide accurate predictions, lowering uncertainty and allowing power system operators to more effectively plan electricity generation and delivery. A dependable power supply may be maintained and operational expenses reduced by precisely anticipating electricity consumption.

iii)The application criteria should be met by the level of detail in load forecasting predictions. Forecasts may be needed on an hourly, daily, or weekly basis, depending on the particular requirements of the system. This level of specificity enables more accurate planning and resource optimisation for power generation and delivery.

iv)Scalable load forecasting methods must be able to handle huge, intricate datasets. The models must be able to correctly forecast demand for various scenarios because the demand for electricity differs between locations and time frames. Scalability guarantees that the models can manage the complexity and volume of data related to various geographic and temporal contexts.

**2.5 Scope of the Work**

The application of deep learning techniques for load forecasting in smart grids is broad and covers a variety of elements. Here are some crucial topics that fall under the purview of this work:

i)Improving load forecasting in smart grids is the main objective. When compared to conventional forecasting techniques, deep learning models have the capacity to capture the complex connections and patterns found in load data. The goal of the study is to improve prediction accuracy by investigating various deep learning architectures, feature engineering methods, and optimisation algorithms.

ii)Real-time forecasting is possible using deep learning techniques, allowing grid operators to make well-informed decisions and improve grid performance. The goal is to create models that can handle streaming data and deliver accurate load projections in a timely manner to assist real-time grid management.

iii)Real-time forecasting is possible using deep learning techniques, allowing grid operators to make well-informed decisions and improve grid performance. The goal is to create models that can handle streaming data and deliver accurate load projections in a timely manner to assist real-time grid management.

v)Uncertainty is a natural part of load forecasting because of a number of variables, including changes in the weather, shifting consumer trends, and unforeseen events. Investigating approaches for quantifying and incorporating uncertainty into deep learning models, like probabilistic forecasting and ensemble methods, in order to produce more accurate and robust load predictions, is part of the scope.

v)Forecasting the amount of traffic on the grid is a key part of energy management and grid optimisation measures. The scope involves incorporating load projections into sophisticated optimisation algorithms for grid stability analysis, demand response, load scheduling, and energy storage management. Energy management systems that are more intelligent and efficient can be developed with the help of deep learning models.

vi)Smart grids require the gathering and analysis of private and sensitive data. Addressing privacy issues and guaranteeing the security of data used for load forecasting are included in the scope. Techniques including data anonymization, encryption, safe data transmission, and adherence to data protection laws could be used for this.

vii)It is within the scope to compare the effectiveness of various deep learning models to more conventional forecasting techniques. In order to determine the best methods for load forecasting in smart grids, it is necessary to assess the proposed models' accuracy, robustness, computing efficiency, and scalability.

**Chapter 3**

**Methodology**

**3.1 Model 1 : LSTM RNN with GRNN**

**i) Recurrent Neural Network(RNN) :**

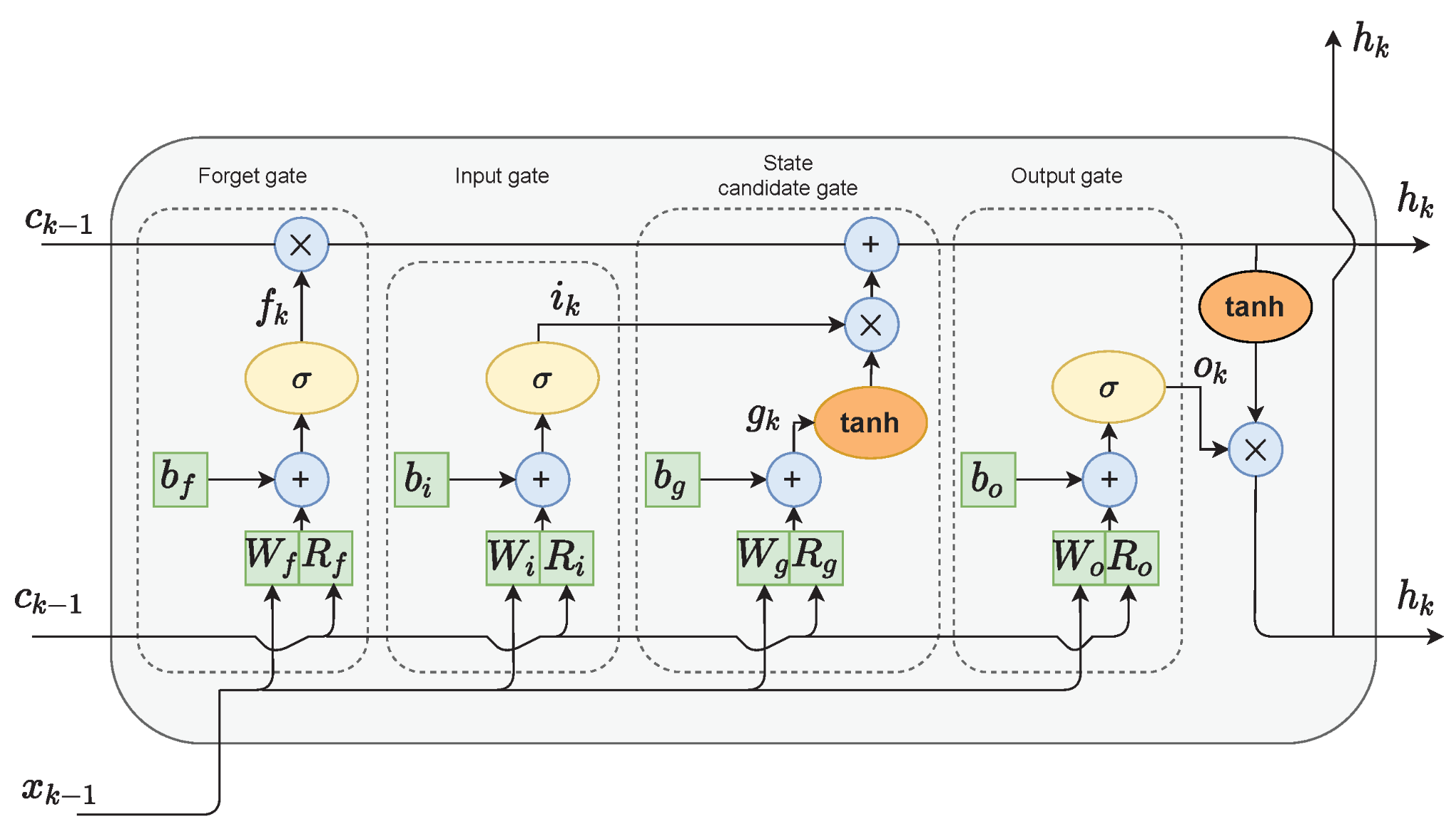
Recurrent neural networks utilize the output from one layer as the input for a subsequent layer. In general, all inputs and outputs are independent of one another, but in the case of RNNs, the idea of memory enters the picture; in order to predict the future, the past must be recalled. It is commonly known that RNN can handle difficulties involving sequence modeling and deal with sequential data.The hidden state, which saves some information about a sequence, is the most crucial aspect of RNNs.RNN is well known to recall information with respect to time and works extremely well with time series forecasting models. This is in contrast to ANN models, which have variable size of neurons, require too much processing, and use non-shared parameters.

However, RNN faces the issue of vanishing gradient, making RNN training particularly challenging.

**ii) Long Short Term Memory(LSTM) :**

Back propagation describes how an error travels from its prediction all the way back to the weights and biases.

Backpropagation through time (BPTT) is the term for this phenomenon, which occurs in recurrent networks such as the RNN and LSTM throughout all time steps despite constant weight and bias matrices.The presence of a gated cell, commonly referred to as the hidden layer, in LSTM architectures marks the primary difference between them and RNN designs. The output and cell state are produced by the interaction of the four internal layers, which make up the system, and are then sent to the neighboring hidden layer. Unlike RNN, LSTM has one tanh layer and three logistic sigmoid gates. Figure 1 shows how the LSTM model looks.



**Fig. 1.** The structure of an LSTM block

**iii) GRNN :**

The supervised FFNN subtype known as the GRNN is one of the most used neural networks.Similar to PNN networks, GRNNs are recognised for their ability to train quickly on sparse data sets. GRNN applications can produce continuous value outputs as compared to PNN's categorization of data.

Data only needs to travel forward once when training GRNN networks, as opposed to most other BPNNs where it may need to travel forward and backward several times until an acceptable error is found.

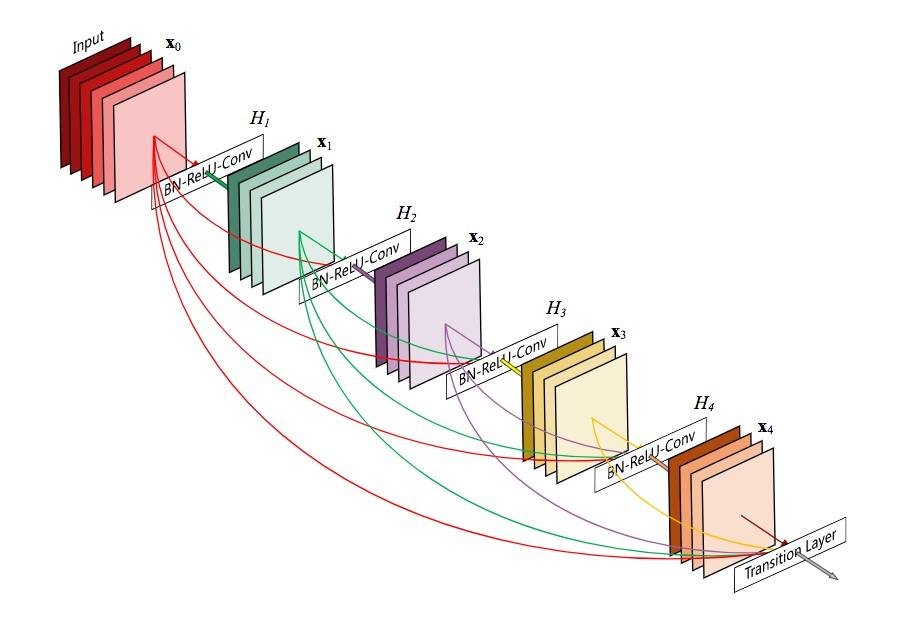
**3.2 Model 2 : DenseNet and ResNet**

**i) DenseNet :**

A DenseNet is a sort of convolutional neural network that makes use of dense connections between layers and Dense Blocks to connect all levels directly to one another. For the sole purpose of reducing the vanishing gradient mistake, DenseNet was created.Information disappears before reaching its destination when the distance between the input and output layers increases.

Every layer in a DenseNet architecture receives input from all the preceding layers and shares its feature-maps with all the succeeding layers.Every layer receives collected data from layers that came before it.

The smaller and more compact form of the network allows for fewer channels.It has greater memory, which makes it more effective at computing.DenseNet enables strong gradient flow, parameter and computational economy, more diverse features, and low complexity features. The network's overall architecture is depicted in the picture below.



**Fig. 2**. The overall architecture of the densely connected

Network

**ii) ResNet :**

ResNet was the organization that first conceptualized the skip connection. They alleviate the problem of disappearing gradients by giving the gradient a different course to pursue. Give the concealed layer an identity function to help it perform as well as the bottom layers. When employing ResNets, gradients from more sophisticated layers to initial filters can travel straight through the skip connections.

**iii) ReLU Activation function**

A common non-linear activation function used in neural networks is the Rectified Linear Unit (ReLU) activation function. It's outlined as:

f(x)=max (0, x)

ReLU produces the maximum of 0 and x for any input x, to put it another way. ReLU sets the output to 0 when the input is negative, thus "turning off" the neuron. ReLU passes an input that is positive through unaltered. ReLU is a frequently used activation function in deep learning because of its effectiveness and simplicity.

**iv) Adam Optimizer**

A well-liked optimisation technique frequently used in deep neural network training is the Adam optimizer. It effectively updates the network weights during training by combining the advantages of two additional optimization techniques, Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).

The acronym "Adam" stands for "Adaptive Moment Estimation," which refers to the estimate of the first and second moments of the gradients as well as the adaptive learning rate. Each parameter has an adjustable learning rate that the algorithm keeps and adjusts throughout training based on the magnitudes of previous gradients. Adam can successfully navigate a variety of data and optimisation environments because of his adaptability.

Because of the Adam optimizer's dependability, effectiveness, and simplicity of use, it has grown in popularity. It handles sparse gradients well, has quick convergence, and is compatible with extensive neural networks. To attain the best performance in various settings, it can be necessary to tune hyperparameters such as learning rate, beta1, beta2, and epsilon.

In conclusion, the Adam optimizer integrates bias correction, momentum, and adjustable learning rates to enhance deep learning network training. It is extensively used because it is good at managing different optimization problems, can speed up convergence, and can enhance model performance.

**3.3 Model 3: SVM using XGDTC**

**i) SVM :**

Each data point in SVM is typically represented as a point in n-dimensional space, and the best line between the two classes of data points is then formed using classification techniques. This line, also known as the hyperplane, is thought to be the most effective one for dividing the data classes. The hyperplane offering the greatest gap between the two classes is selected. It is occasionally also referred to as the decision border.

Without the requirement for extensive data transformations, SVMs are utilized to manage complex relationships. For both small and complicated datasets, they are able to produce accurate findings.

SVM may provide both pre-made and bespoke kernels for the decision functions, is memory-efficient, and performs well in high-dimension situations.

**ii) Decision Tree Classifier :**

The training model is created using a Decision Tree, and it may be used to forecast the class or value of the target variable by adopting straightforward decision-making rules obtained from historical data.They can be categorical or continuous, and they often work with the idea of nodes, branches, and splitting using judgment. They feature a hierarchical tree structure and are non parametric.

Using loss functions, decision trees assess the split according to purity. They can handle both continuous and categorical values and require less computation for categorization. They can also point out which fields are crucial for obtaining precise predictions or classifications.

**iii) Gini Index :**

The Gini Index, which is a number, is used to assess a split's accuracy among the groups that were classified. The Gini index is used to assess a score between 0 and 1, where 1 denotes a randomly distributed distribution of the items within classes and 0 denotes that all observations are members of the same class.

Decision trees enable data robustness, interpretability, and minimal processing.

**iv) XG-Boost :**

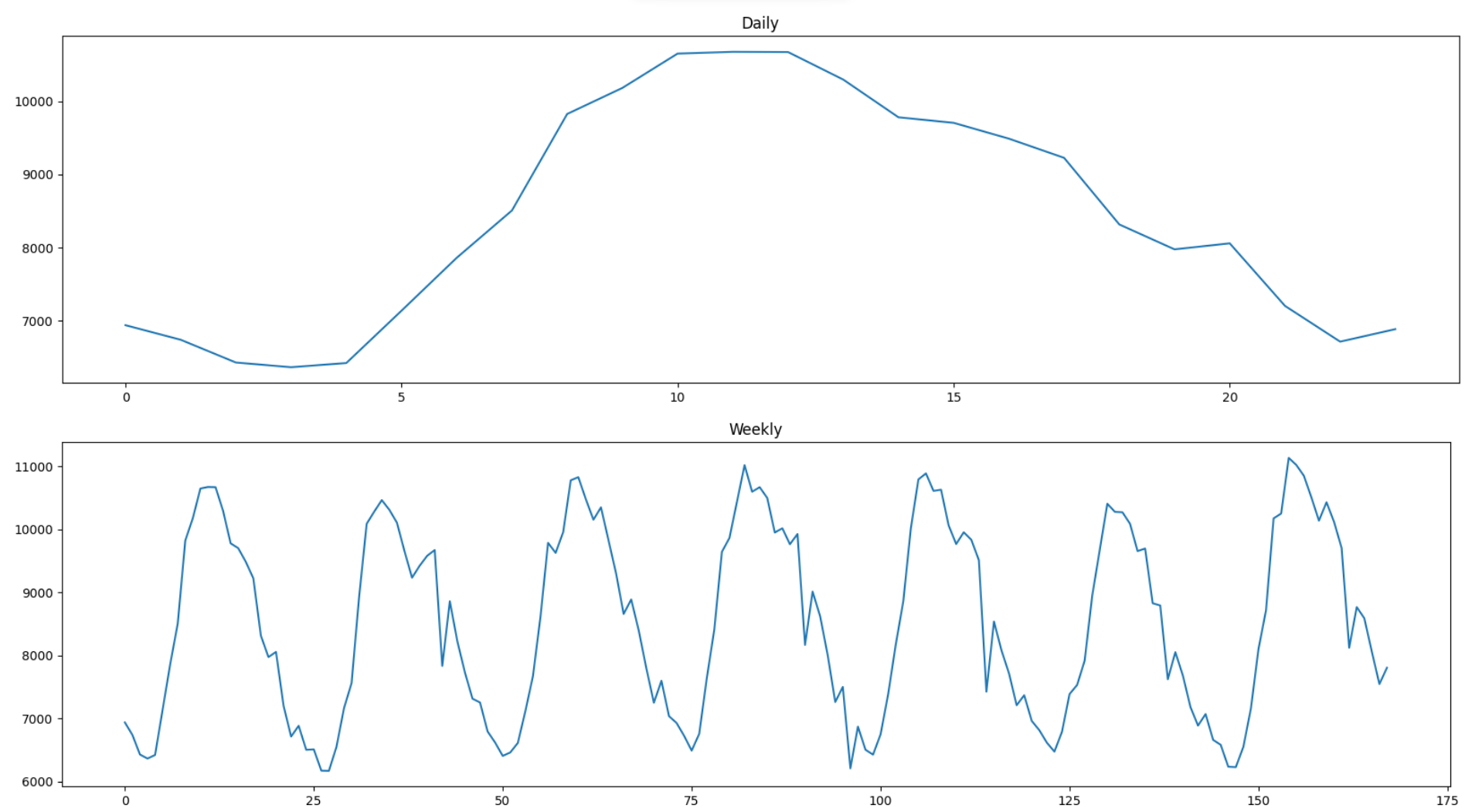
XGBoost is a high-performance gradient boosted trees solution created for supervised learning problems that demand precise target variable prediction. It makes use of an ensemble machine learning technique that combines the predictions of several weaker, simpler models and enhances the precision of the predictions through the use of a gradient boosting framework. In order to get better results faster, XGBoost has a decision-tree based structure and is hardware and software optimized.

**Chapter 4**

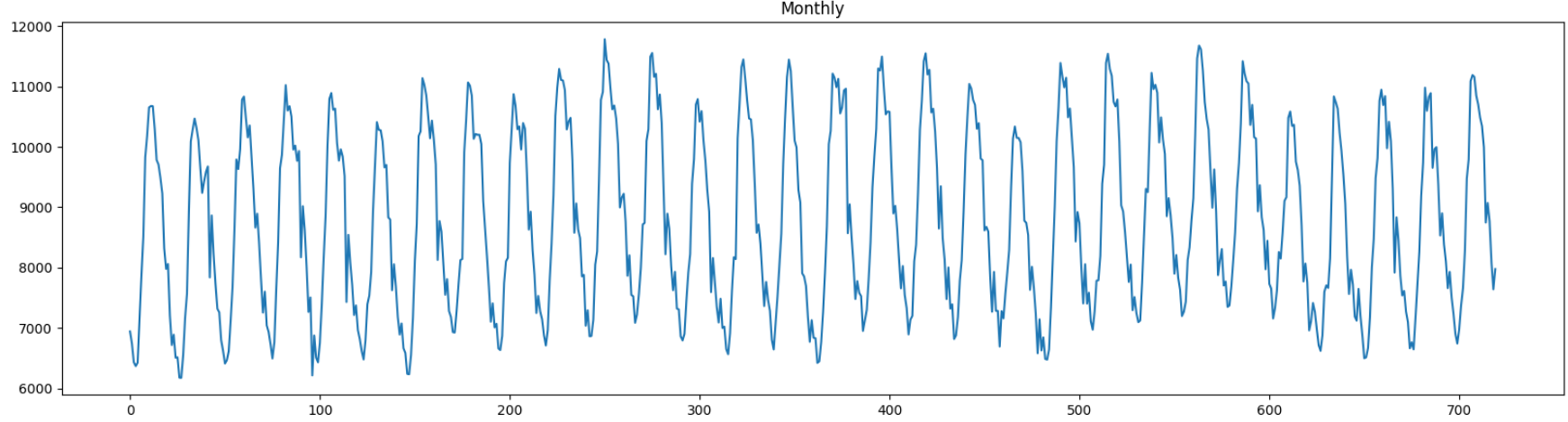
**Implementation Details**

**4.1 LSTM RNN Model**

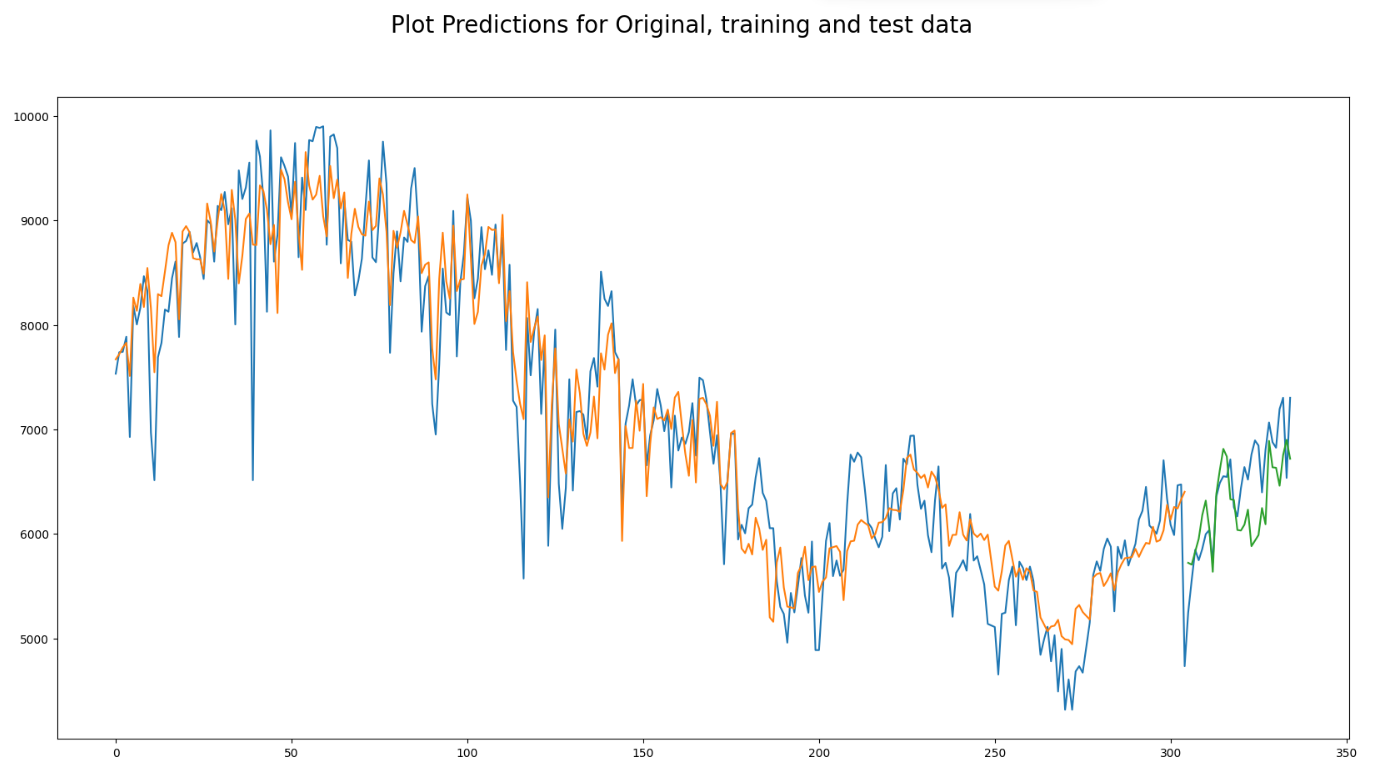
The model has been implemented by the Karnataka Load Dispatch Center for the year 2019 which contains hourly load demand for every day across 12 months consisting of 8760 entries. The dataset was preprocessed and the RNN-LSTM algorithm was applied, where the LSTM NN's input array consists of matrix cells, two popular statistical metrics, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used to evaluate the forecasting algorithms.



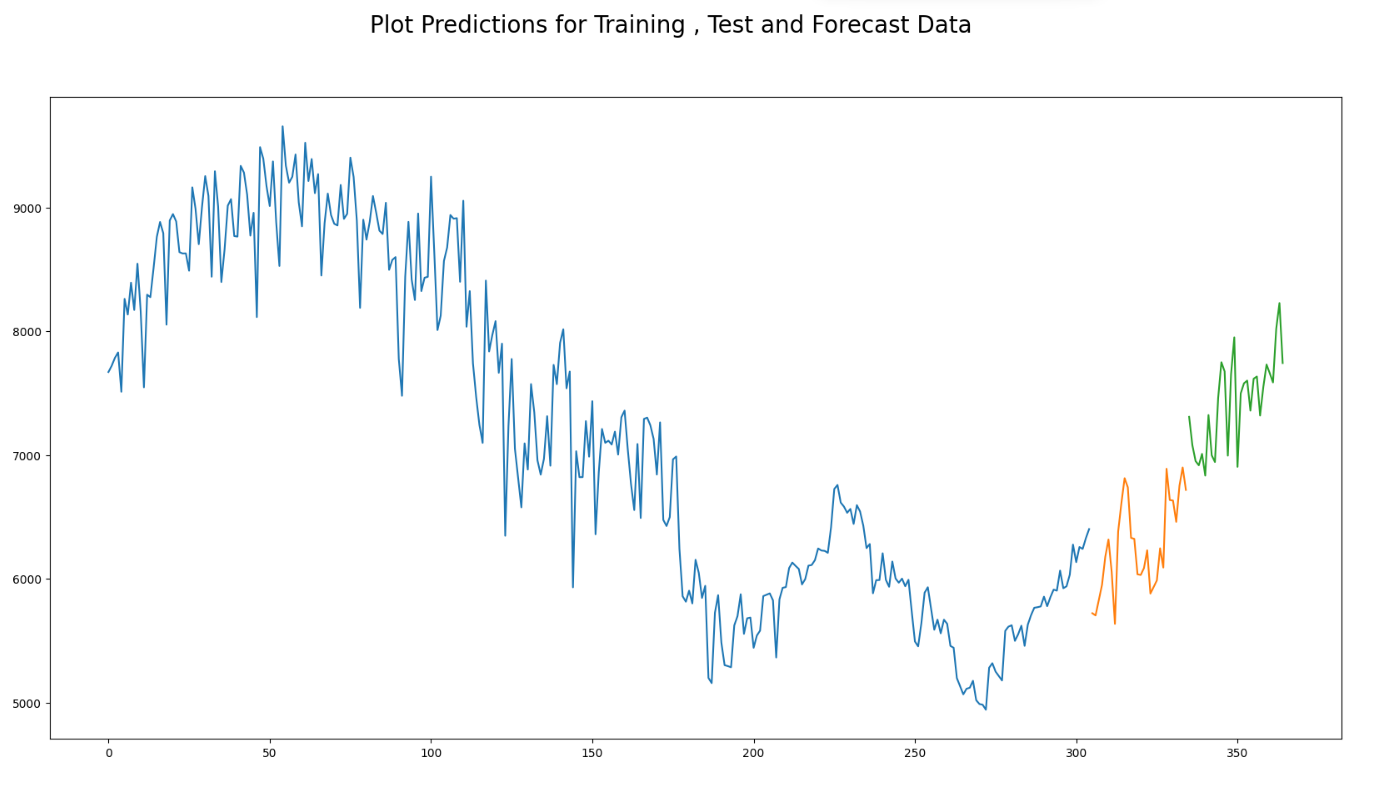
**Fig. 3.** Daily and Weekly load demand analysis



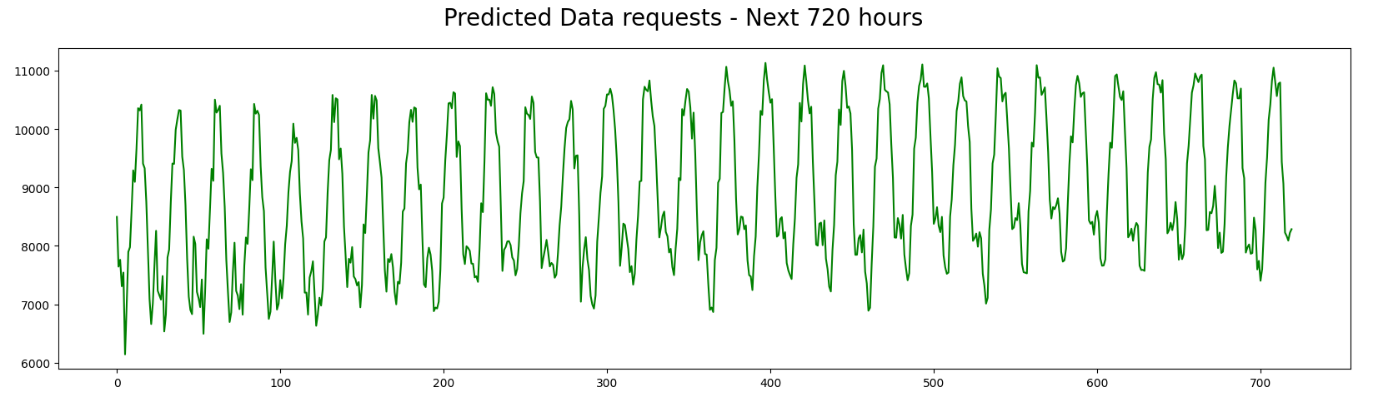
**Fig. 4.** Monthly Load Analysis



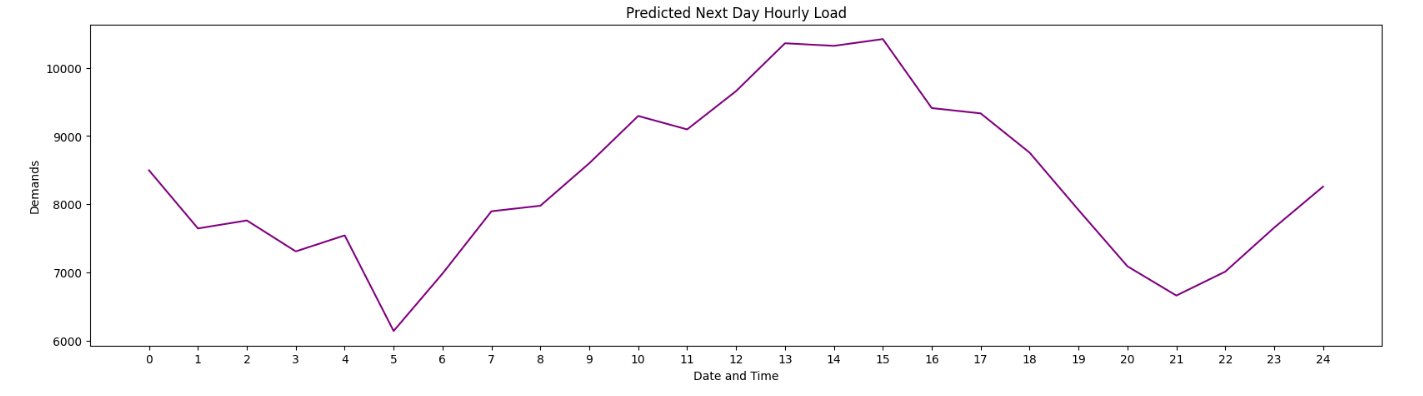
**Fig. 5.** Plot Predictions for Original, Training and Test Data



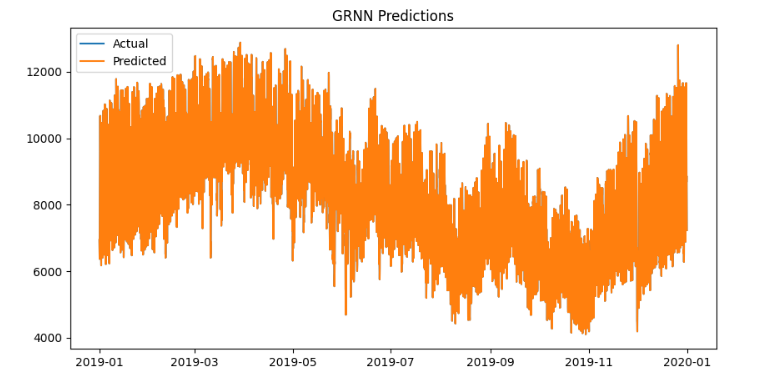
**Fig. 6.** Plot Predictions for training, test and forecast data



**Fig. 7.** Predicted Data requests for the next 720 hours



**Fig. 8.** Predicted Data requests for the day on hourly basis



**Fig. 9.** GRNN predictions

TABLE IV.

RMSE, MAE AND ACCURACY OF MODEL

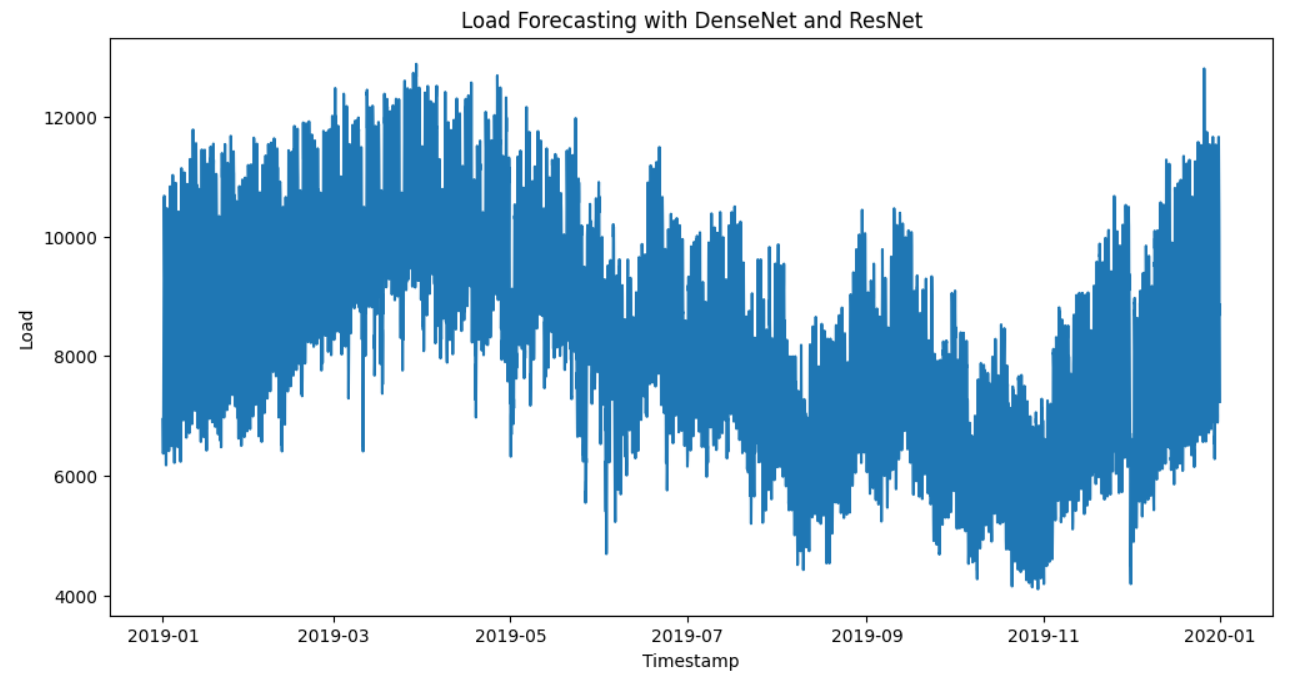
| Model | RMSE | MAE | Accuracy(%) |
| --- | --- | --- | --- |
| LSTM | 20.51 | 27.41 | 72.58 |
| RNN | 20.42 | 27.27 | 72.72 |

**4.2 DenseNet and ResNet Model**

The model is implemented using the same dataset. We used a model framework influenced by Residual Network and DenseNet. 200 neurons make up each layer of the model, which are added together at the summation layer. By utilizing all of the prior connections, it functions like a DenseNet internal block.

Instead of concatenating the layers, they are summed up like the residual connection. Compared to the DenseNet structure, the model has less connections. According to its speed and accuracy, each layer in the model is a Dense layer employed with the ReLU Activation function and Adam Optimizer.

Using k-folds validation (where folds=5), the models were implemented on base, undervalued and overvalued datasets. The model has achieved an accuracy of 94.79% (base dataset), 98.82% (under sampling) and 99.01% (over sampling).



**Fig. 10.** Predicted load using DenseNet and ResNet

Table V.

RMSE, MAE AND ACCURACY OF MODEL

| Model | RMSE | MAE | Accuracy(%) |
| --- | --- | --- | --- |
| DenseNet | 15.02 | 12.10 | 99 |
| ResNet | 16.49 | 13.20 | 99 |

Table VI.

ACCURACY OF BASE, OVERSAMPLED & UNDERSAMPLED DATASET

| Base | Over Sampled | Under Sampled |
| --- | --- | --- |
| 94.79% | 98.82% | 99.01% |

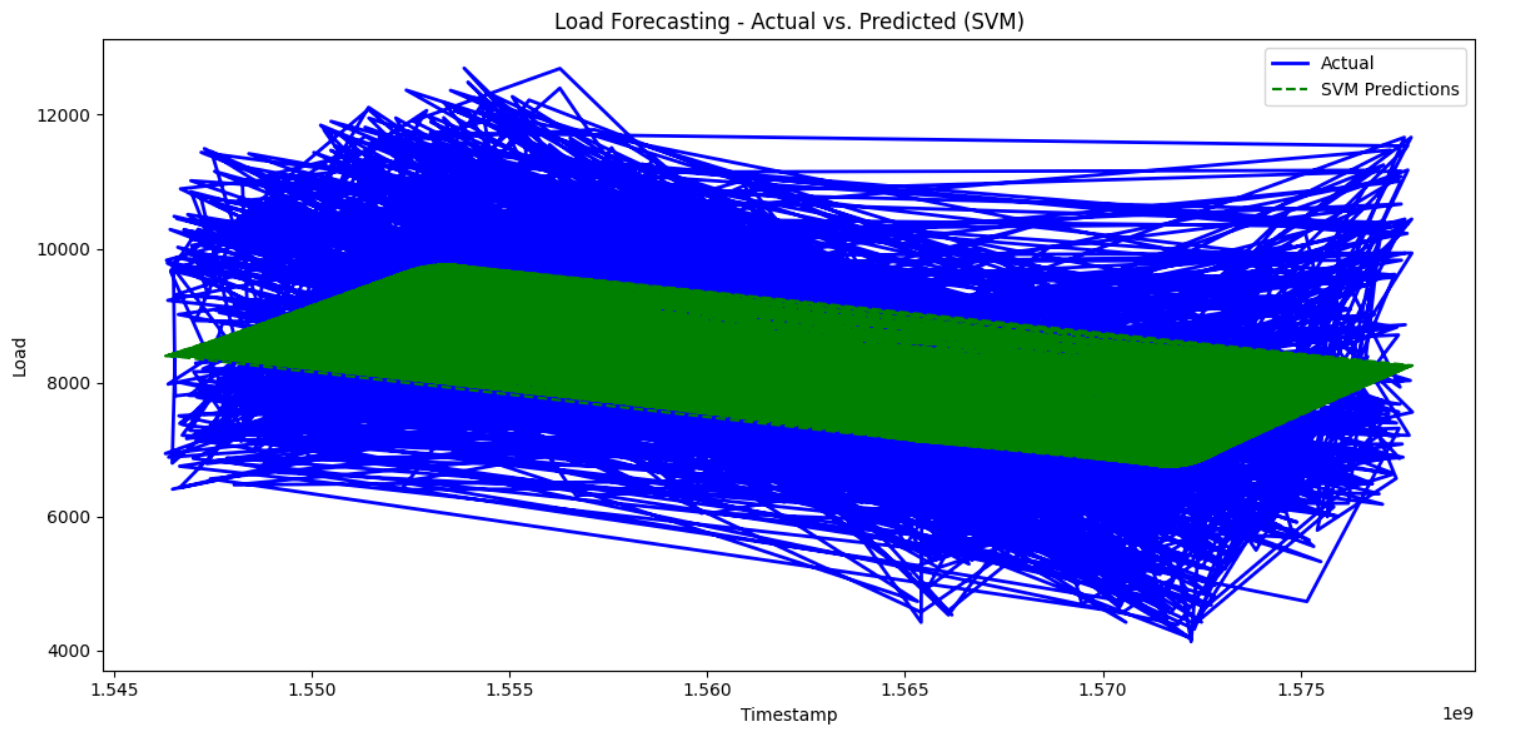
**4.3 SVM and XG-DTC Model**

The model is implemented using the same dataset. After data processing, it is made available to the SVM , through which the data passes through the tuning parameters. Validation data is used to test the first network that is constructed. The forecasting network retrains and fine-tunes itself using validation data after receiving the forecasted results until the forecasting errors are at their lowest possible level. In the end, load forecasting uses the final, configured network measurements to evaluate the forecasting performance.

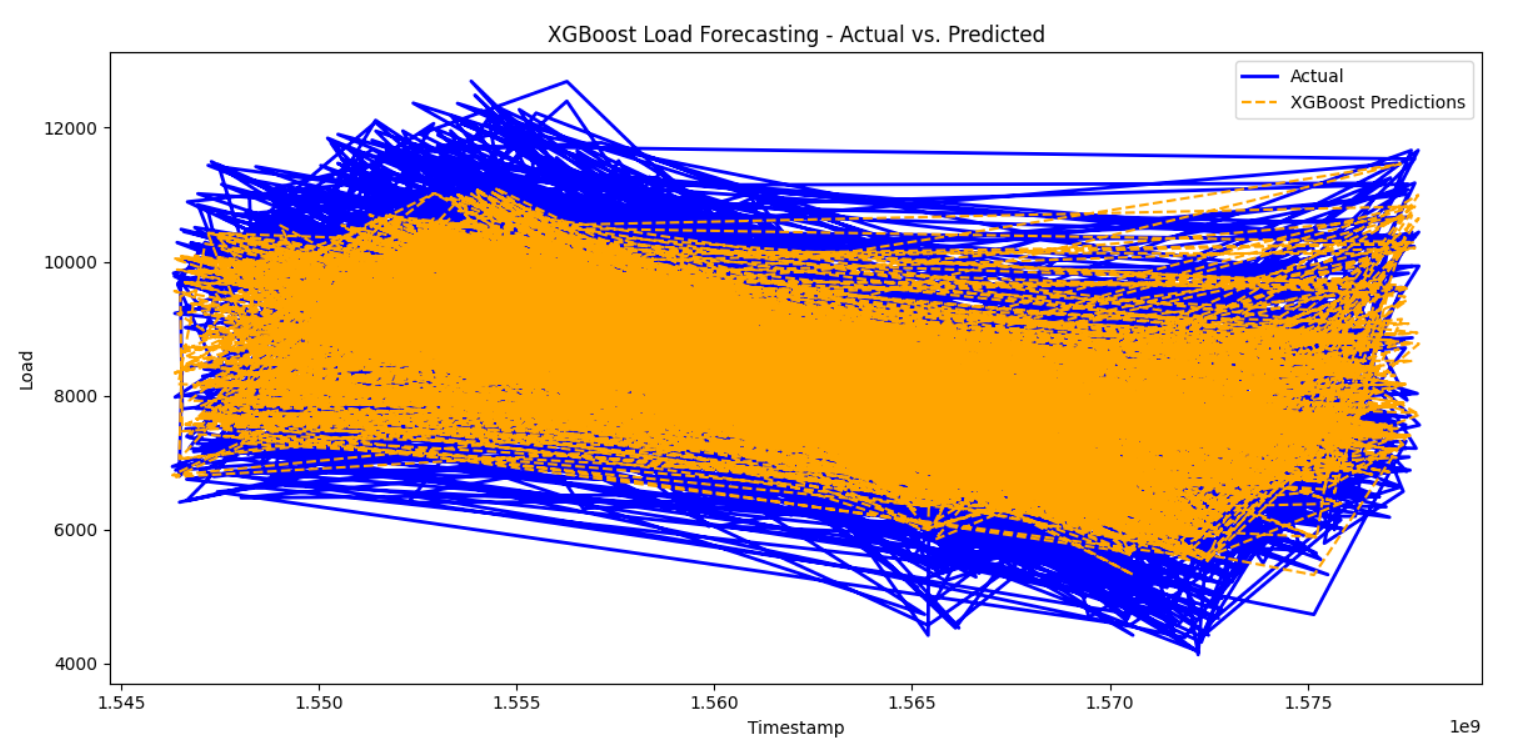
In order to choose the best features in the input data, a mixture of the two strategies (XGBoost and DTC) is used for feature selection. We implemented a network with a hybrid feature selector, extraction based on RFE, and classifier based on SVM to boost precision.

On the basis of test data, the network is evaluated, and weekly and monthly load forecasts RMSE and MAPE metrics are calculated to check the accuracy of the model.

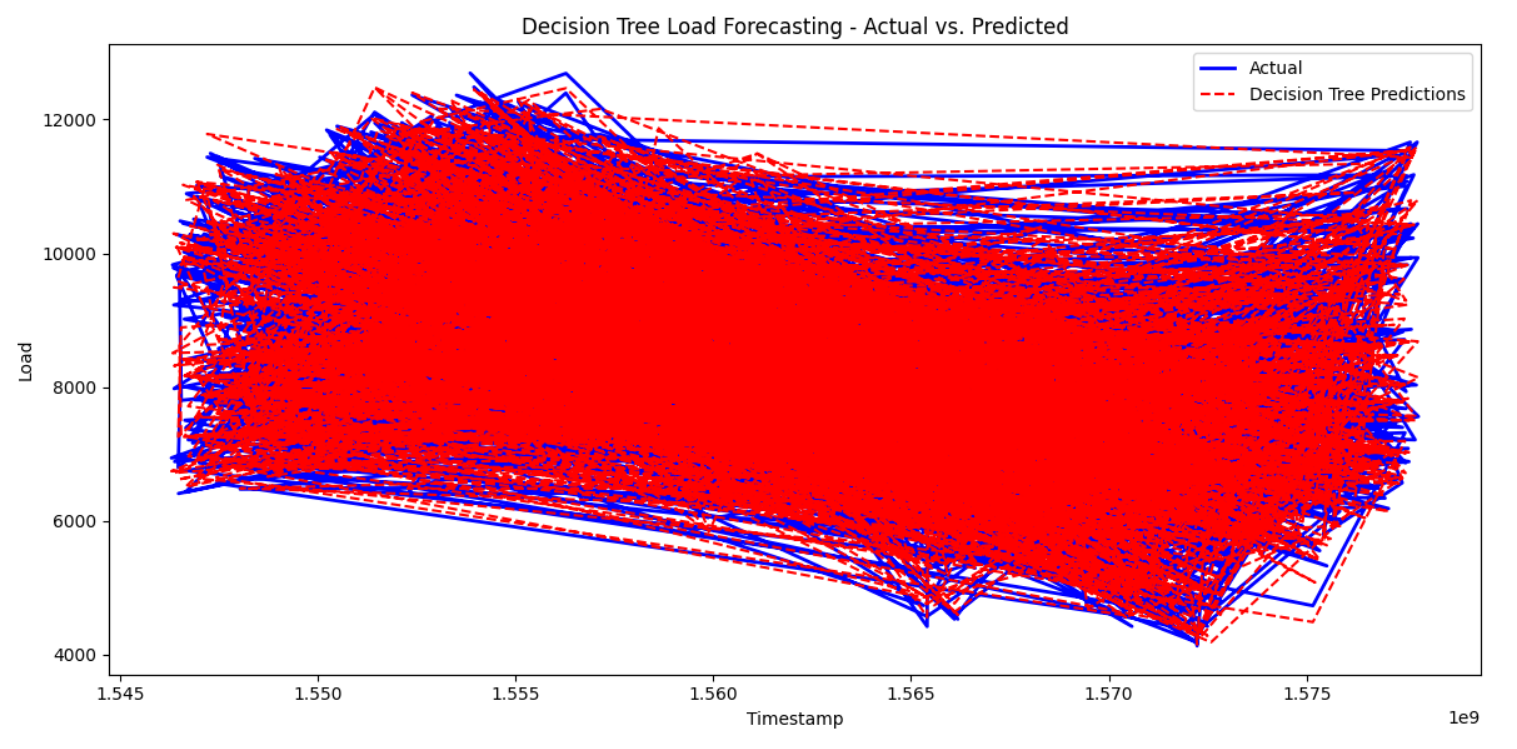
Since the SVM XG-DTC is a regression based model, accuracy of the model could not be calculated.



**Fig. 11.** SVM Load forecasting - Actual vs. Predicted Data



**Fig. 12.** XG-Boost Load forecasting - Actual vs. Predicted Data



**Fig. 13.** Decision Tree Load forecasting - Actual vs. Predicted Data

TABLE VII.

RMSE & MAE OF THE MODEL

| Model | RMSE | MAE |
| --- | --- | --- |
| SVM | 12.88 | 10.57 |
| XG-Boost | 8.86 | 7.49 |
| DTC | 5.11 | 3.85 |

**4.4 Software Used**

This project is developed using Python 3.8.8 , PyCharm IDE, Google Colab Notebooks and Google GPU & TPU.

The libraries used in this project:

1. NumPy: NumPy, aka "Numerical Python," is an extremely popular open-source Python library. It provides extensive support for handling large multi-dimensional arrays and matrices, along with a wide range of mathematical operations that can be performed on them. Due to its efficient array operations and mathematical capabilities, NumPy serves as a fundamental library in scientific computing, data analysis, and machine learning applications.

2. Matplotlib: Matplotlib is a widely-used open-source Python library that enables the creation of static, animated, and interactive visualizations. It offers a versatile and extensive set of tools for generating plots, charts, and various graphical depictions of data.

3. SciPy: SciPy is a Python library dedicated to technical and scientific computing, which serves as an open-source solution. It builds upon the capabilities of NumPy and extends them by incorporating additional features for optimization, integration, interpolation, linear algebra, signal processing, statistics, and more. SciPy offers an extensive collection of numerical and scientific computing routines, encompassing tasks such as numerical integration, optimization, linear algebra, interpolation, and Fourier analysis. These functions are carefully optimized for performance, ensuring efficient implementations for intricate scientific calculations.

4. TensorFlow: TensorFlow is a deep learning framework that has been developed by Google as an open-source project. It offers a flexible and efficient environment for constructing, training, and deploying machine learning models, with a specific emphasis on deep neural networks. This framework provides a powerful platform that enables users to work with various aspects of machine learning, empowering them to build.

5. Pandas: Pandas is an open-source Python library known for its efficient data manipulation and analysis capabilities. It provides specialized data structures like Series and DataFrame, which allow for flexible handling of data. Pandas is widely used for tasks such as data cleaning, filtering, sorting, grouping, merging, and reshaping. With its rich functionalities, Pandas facilitates data preparation, exploration, and analysis, making it a powerful tool for data wrangling.

6. Seaborn: Seaborn is a Python library for data visualization that operates as an open-source solution, built upon the foundation of Matplotlib. It offers a high-level interface that enables the creation of visually appealing and informative statistical graphics. Seaborn is specifically designed to simplify the visualization of complex datasets, providing convenient features for statistical estimation and color palettes. With Seaborn, users can effortlessly generate visually appealing and insightful visualizations for effective data exploration and communication.

7. Keras: Python-based Keras is an open-source framework for deep learning. For creating, honing, and deploying deep neural networks, it offers a high-level, user-friendly interface. Keras is designed to be easy to use, efficient, and extensible, making it a popular choice for both beginners and experienced practitioners in the field of deep learning. Keras offers a simple and intuitive API that allows users to define and train deep neural networks with just a few lines of code. It provides a high-level abstraction for building complex neural networks, making it easy to experiment with different network architectures and hyperparameters

8. Scikit-Learn :A well-liked open-source Python machine learning toolkit called Scikit-learn offers a variety of tools and methods for various machine learning applications, including model selection, regression, clustering, classification, and preprocessing. It is a strong and complete tool for machine learning tasks since it is built on top of other scientific libraries like NumPy, SciPy, and matplotlib.

Due to Scikit-learn's efficiency, adaptability, and simplicity, it is extensively utilized in both academia and business. It is a useful tool for data scientists, academics, and practitioners in the field of machine learning since it enables users to easily develop and experiment with machine learning algorithms.

9. Google Colab : Users may write and run Python code in a browser using Google Colab, the abbreviation for Google Colaboratory, a cloud-based development environment offered by Google. It offers a free computing environment with GPU capability on top of the Jupyter Notebook.

Due to its simplicity of use, accessibility, and affordability, Google Colab is a preferred option for new users, academics, and data scientists. It provides a practical environment for developing and running code, particularly for projects combining machine learning, data analysis, and teamwork.

10. Google GPU : The Google Cloud Platform (GCP), a cloud computing platform, offers GPU (Graphics Processing Unit) resources. Users may take use of the numerous GPU choices provided by GCP for a variety of computing activities, including deep learning, scientific simulations, and data processing.

Users may access strong processing resources for GPU-accelerated applications with Google GPU on Google Cloud Platform. It enables academics, data scientists, and developers to take advantage of GPUs' parallel processing capabilities to accelerate complicated computations and effectively handle computationally demanding jobs.

11. Google TPU: Tensor Processing Unit, or Google TPU, is an application-specific integrated circuit (ASIC) that Google constructed specifically to speed up machine learning workloads. Deep learning workloads are especially optimized for TPUs, which offer high-performance, low-latency processing.

A specialized piece of technology for speeding up deep learning tasks is Google TPU. It is a useful tool for academics, data scientists, and developers working on computationally challenging machine learning tasks due to its optimized architecture and high-performance features. Users may employ TPUs to speed up their training and inference processes, allowing for quicker iteration and increased productivity in the deep learning space.

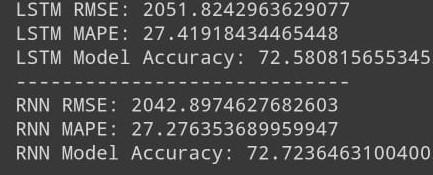
**Chapter 5**

**Result and Analysis**

In this section we present the results of the load forecasting models and discuss their performance based on the evaluation metrics, namely Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and overall accuracy.

**Model 1:**

The RNN-LSTM model exhibits promising performance in load forecasting. RMSE values of 20.42 and 20.51 are obtained for the RNN and LSTM models respectively, along with MAE values of 27.27 and 27.41 with accuracy levels of 72.72% and 72.58% respectively.

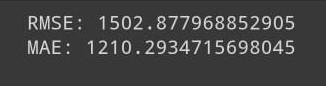


**Fig.14.** RMSE,MAE and Accuracy for RNN and LSTM

The plot of the predicted load values against the actual load values demonstrates a close alignment, indicating a good fit between the predicted and actual data points. The time series trend and patterns are well-captured, validating the model's ability to capture temporal dependencies and long-term load patterns.

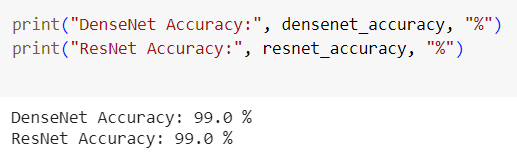
**Model 2:**

The DenseNet model also demonstrates competitive performance in load forecasting. RMSE values of 15.02 and 16.49 are obtained for the DenseNet and ResNet models respectively, along with MAE values of 12.10 and 13.20 with accuracy levels of 99% each respectively.



**Fig.15.** RMSE AND MAE Values for DenseNet and ResNet.

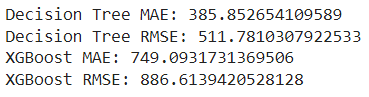
The plot comparing the predicted and actual load values reveals a close alignment between the two, indicating that the model effectively captures the underlying load patterns. Although there may be some deviations, the overall trend and major load fluctuations are well-predicted by the model.



**Fig.16.** RMSE AND MAE Accuracy Values for DenseNet and ResNet.

**Model 3:**

The hybrid model combining SVM, XGBoost, and DTC exhibits competitive performance in load forecasting. The computed RMSE value for SVM is 91.80, 54.65 for XG-Boost and 19.20 for DTC. The MAE value is 68.36 for SVM, 43.64 for XG-Boost and 14.96 for DTC. The plot comparing the predicted and actual load values demonstrates a close alignment, indicating that the hybrid model effectively captures the underlying load patterns. The model's combination of SVM, XGBoost, and DTC enables accurate load demand predictions by leveraging the strengths of each component.





**Fig.17.** RMSE AND MAE Values for SVM,DTC and XgBoost

**Chapter 6**

**Conclusion and Future Scope**

In this project, we introduced three distinct deep learning models for short and mid-term load forecasting using various standard datasets. The paper compares and contrasts the three models, examining their respective methods, definitions, and implementations. Overall, all models—RNN-LSTM, DenseNet, ResNet, and the hybrid SVM-XGBoost-DTC—exhibit competitive performance in load forecasting, showcasing their suitability for smart grid applications. While slight differences in RMSE, MAPE, and accuracy levels are observed, all models demonstrate accurate load demand predictions and capture the underlying load patterns effectively.

The selection of a specific model for a given application can depend on various factors, including the specific requirements, computational efficiency, and interpretability. Researchers and practitioners can leverage the presented results and plots to make informed decisions regarding the choice of load forecasting model in smart grid environments.

The analysis of the literature revealed that load forecasting is crucial for maximizing energy production, distribution, and consumption. Numerous academic articles and unique studies emphasized the use of deep learning methods in obtaining precise load forecasts. These methods make use of neural network technology to identify intricate relationships and patterns in data on energy use.

The project report was organized logically, starting with an introduction that included a summary of the goals of the study and the significance of load forecasting in smart grids. The deep learning load forecasting studies and methodology were examined in the literature review.

The deep learning models employed, such as LSTM, DenseNet, and ResNet, which have been shown to be successful in capturing temporal relationships and extracting useful features, were detailed in the technique section. To take advantage of the advantages of ensemble learning, a combination model comprising SVM, XGBoost, and Decision Tree Classifier was used.

The complexity of load forecasting, such as data quality, feature selection, model selection, and interpretability, was highlighted in the section on outstanding topics and difficulties. These difficulties provide possibilities for more study and development in the area of load forecasting in smart grids.

The requirement for developing precise load forecasting models that can manage the dynamic and unpredictable nature of energy use in smart grids was emphasized in the issue description. The project's goal was to solve this issue by utilizing deep learning techniques, which have demonstrated promising outcomes in capturing the intricate correlations seen in data on energy use.

The use of ensemble models and well-known deep learning methods demonstrated the adaptability and flexibility of the project's methodology. The project aims to improve load forecasting accuracy and contribute to the effective management of energy resources in smart grids by utilizing the capabilities of deep learning and merging various models.

This study investigated the use of deep learning techniques for load forecasting in smart grids. The application of LSTM, DenseNet, ResNet, and ensemble models demonstrated the possibility of increased load forecast accuracy. The experiment clarified the significance of precise load forecasting in smart grids and the possibility of deep learning approaches to overcome this difficulty.

The outcomes and learnings from this project may be used as a basis for further study and improvements in load forecasting techniques, which will ultimately result in smart grid energy management that is more effective and sustainable.

The potential application of deep learning techniques in load forecasting for smart grids spans a number of research and development areas:

1. To increase accuracy and interpretability, investigating the pairing of deep learning with other forecasting methodologies.
2. By developing methods to make deep learning models for load forecasting more interpretable, stakeholders will be better able to comprehend the variables affecting load projections.
3. Investigating techniques to apply knowledge gained from one smart grid or dataset to another with similar properties, enabling more accurate predictions. Transfer Learning and Domain Adaptation.
4. Integration of External elements: Extending deep learning models to take into account a wider range of external elements, such as socioeconomic data, infrastructural changes, and events, that affect load patterns.
5. Using edge computing and federated learning approaches, load forecasting models may be trained and deployed at the grid's edge, protecting data privacy and minimizing reliance on centralized infrastructure.
6. Adaptive and Self-Learning Models: Creating self-learning models that are constantly improving their predicting abilities and adaptive models that can automatically adjust to changing conditions.

# Bibliography:

[1] S. Ryu, J. Noh, and H. Kim, "Deep neural network based demand side short term load forecasting," in 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2016, pp. 308-313.

[2] R. Zhang, Z. Y. Dong, Y. Xu, K. Meng, and K. P. Wong, "Short-term load forecasting of Australian National Electricity Market by an ensemble model of extreme learning machine," IET Generation, Transmission & Distribution, vol. 7, pp. 391-397, 2013.

[3] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128-3137.

[4] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-RNN)," arXiv preprint arXiv:1412.6632, 2014.

[5] A. Graves and N. Jaitly, "Towards End-To-End Speech Recognition with Recurrent Neural Networks," in ICML, 2014, pp. 1764-1772.

[6] X. Cao, S. Dong, Z. Wu, and Y. Jing, "A Data-Driven Hybrid Optimization Model for Short-Term Residential Load Forecasting," in Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, 2015, pp. 283-287. [7] Z. Yun, Z. Quan, S. Caixin, L. Shaolan, L. Yuming, and S. Yang, "RBF Neural Network and ANFIS-Based Short-Term Load Forecasting Approach in Real-Time Price Environment," IEEE Transactions on Power Systems, vol. 23, pp. 853-858, 2008.

[8] H. Li, Y. Zhao, Z. Zhang, and X. Hu, "Short-term load forecasting based on the grid method and the time series fuzzy load forecasting method," in International Conference on Renewable Power Generation (RPG 2015), 2015, pp. 1-6.

[9] P. Qingle and Z. Min, "Very Short-Term Load Forecasting Based on Neural Network and Rough Set," in Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on, 2010, pp. 1132-1135.

[10] R. Zhang, Z. Y. Dong, Y. Xu, K. Meng, and K. P. Wong, "Short-term load forecasting of Australian National Electricity Market by an ensemble model of extreme learning machine," IET Generation, Transmission & Distribution, vol. 7, pp. 391-397, 2013.

[11] R. Zhang, Y. Xu, Z. Y. Dong, and W. Kong, "A Composite k-Nearest Neighbor Model for Day-Ahead Load Forecasting with Limited Temperature Forecasts," presented at the IEEE General Meeting Boston, 2016.

[12] F. H. Al-Qahtani and S. F. Crone, "Multivariate k-nearest neighbor regression for time series data - A novel algorithm for forecasting UK electricity demand," in Neural Networks (IJCNN), The 2013 International Joint Conference on, 2013, pp. 1-8.

[13] M. Ghofrani, M. Ghayekhloo, A. Arabali, and A. Ghayekhloo, "A hybrid short-term load forecasting with a new input selection framework," Energy, vol. 81, pp. 777-786, 3/1/ 2015.

[14] P. Zhang, X. Wu, X. Wang, and S. Bi, "Short-term load forecasting based on big data technologies," CSEE Journal of Power and Energy Systems, vol. 1, pp. 59-67, 2015.

[15] F. L. Quilumba, W. J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, "Using Smart Meter Data to Improve the Accuracy of Intraday Load Forecasting Considering Customer Behavior Similarities," IEEE Transactions on Smart Grid, vol. 6, pp. 911-918, 2015.

[16] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating Practice Theory in Sub-Profile Models for Short Term Aggregated Residential Load Forecasting," IEEE Transactions on Smart Grid, vol. PP, pp. 1-8, 2016.

[17] M. Chaouch, "Clustering-Based Improvement of Nonparametric Functional Time Series Forecasting: Application to Intra-Day Household-Level Load Curves," IEEE Transactions on Smart Grid, vol. 5, pp. 411-419, 2014.

[18] M. Ghofrani, M. Hassanzadeh, M. Etezadi-Amoli, and M. S. Fadali, "Smart meter based short-term load forecasting for residential customers," in North American Power Symposium (NAPS), 2011.

[19] S. Ryu, J. Noh, and H. Kim, "Deep neural network based demand side short term load forecasting," in 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2016, pp. 308-313.

[20] Hossain, M. S., & Mahmood, H. (2020). Short-Term Load Forecasting Using an LSTM Neural Network. 2020 IEEE Power and Energy Conference at Illinois (PECI).

[21]Dataset:https://ieee-dataport.org/documents/data-set-used-conferencepaper-titled-short-term-load-forecasting-using-lstm-neural#files