TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A THESIS REPORT ON SHORT-TERM ELECTRICAL LOAD FORECASTING FOR BANESHWOR FEEDER USING MACHINE AND DEEP LEARNING MODELS

SUBMITTED BY:

SUJIT KOIRALA
(PUL075MSPSE016)

SUPERVISED BY:

PROF. DR. ROGERS S PRESSMAN

SUBMITTED TO:

DEPARTMENT OF ELECTRICAL ENGINEERING

December, 2025

Declaration

I hereby declare that this study/research entitled [**Put the title of the project/thesis here....**] is based on our original research work. Related works on the topic by other researchers have been duly acknowledged. I owe all the liabilities relating to the accuracy and authenticity of the data and any other information included hereunder.

Name of the Student (Roll no)

Date:

Recommendation

This is to certify that this project report entitled [**Put the title of the project/thesis here**] prepared and submitted by [**Put the name & roll no of the student here**], in partial fulfillment of the requirements of the Master degree of Engineering in......, awarded by Tribhuvan University, has been completed under my/our supervision. I/we recommend the same for acceptance by Tribhuvan University.

Name of the Supervisor: ABC
Designation: ABC
Organization: ABC
Date: ABC

Name of the Co-supervisor: ABC
Designation: ABC
Organization: ABC
Date: ABC

Page of Approval

TRIBHUVAN UNIVERSIY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

This project/thesis entitled [**Put the title of the project/thesis here**] prepared and submitted by [**Put the Name and Roll no of the student here**] has been examined by us and is accepted for the award of the Master's degree in [Put name of the program] by Tribhuvan University.

....	....
Supervisor	External examiner
Name	**Name**
Designation	Designation
Organization Name and Address.	Organization Name and Address.

....
Co-Supervisor (if Any)
Name
Designation
Organization Name and Address.

Date of approval:

Copyright

Head
Department of Electronics and Computer Engineering
Pulchowk Campus, Institute of Engineering, TU
Lalitpur, Nepal.

Acknowledgments

I would like to express my sincere gratitude to my supervisor and faculty members of the Department of Electrical Engineering for their valuable guidance, continuous support, and encouragement throughout the course of this project. Their technical insights and constructive feedback were instrumental in shaping this work. I am also thankful to the Nepal Electricity Authority and relevant data-providing institutions for making the load and meteorological data available for this study. Their cooperation greatly contributed to the successful completion of the analysis. Special thanks go to my friends and colleagues for their support, discussions, and motivation during the project period. Finally, I would like to express my heartfelt appreciation to my family for their constant encouragement and support throughout my academic journey.

Sujit Koirala (PUL075MSPSE016)

Abstract

Accurate short-term electrical load forecasting plays a crucial role in the efficient planning and operation of modern power systems. With increasing load variability influenced by weather conditions, temporal patterns, and socio-economic activities, traditional statistical methods often struggle to capture complex and nonlinear demand behavior. This project focuses on short-term electrical load forecasting for the Lekhnath Feeder using machine learning–based approaches.

Historical hourly load data, along with meteorological variables such as air temperature, global solar radiation, and relative humidity, were used to develop predictive models. Comprehensive data preprocessing was performed, including missing value imputation, outlier treatment, temporal feature extraction, and cyclical encoding of time-based variables. Several machine learning models were implemented and evaluated, including Linear Regression, Ridge Regression, Support Vector Regression, Random Forest, Gradient Boosting, and XGBoost. Hyperparameter tuning was applied to improve model performance.

The models were assessed using standard evaluation metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2). The results show that ensemble-based models, particularly tuned XGBoost and Random Forest models, significantly outperform linear and baseline methods. The findings highlight the effectiveness of machine learning techniques for feeder-level short-term load forecasting and provide valuable insights for operational planning and decision-making in power distribution systems.

Keywords: *short-term load forecasting, machine learning, XGBoost, Random Forest, power distribution systems*

Contents

List of Figures

List of Tables

List of Abbreviations

NEA	Nepal Electricity Authority
STLF	Short-Term Load Forecasting
ML	Machine Learning
DL	Deep Learning
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
MLP	Multi-Layer Perceptron
SVR	Support Vector Regression
RF	Random Forest
GBR	Gradient Boosting Regressor
XGBoost	Extreme Gradient Boosting
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
SMAPE	Symmetric Mean Absolute Percentage Error
$\mathbf{R^2}$	Coefficient of Determination
MW	Megawatt
BS	Bikram Sambat (Nepali Calendar)
AD	Anno Domini (Gregorian Calendar)
EDA	Exploratory Data Analysis
IQR	Interquartile Range
TFT	Temporal Fusion Transformer
API	Application Programming Interface

List of units and conversions

m^3	Meter cube (Cubic meter)
Sq.ft	Square feet
add	more

1. Introduction

This project focuses on short-term electrical load forecasting at the feeder level using data-driven machine learning and deep learning techniques. Historical load data combined with weather and temporal features were used to model and predict hourly power demand. Multiple forecasting models were developed and evaluated to identify the most effective approach for accurate and reliable load prediction.

1.1 Background

Electricity demand is never constant. It rises and falls with daily routines, temperature changes, business hours, and countless other factors. For a power system operator, being able to predict this demand even just a few hours ahead can make a huge difference. Accurate short-term forecasting helps optimize generation schedules, reduce operational costs, manage peak hours more confidently, and maintain a reliable supply.

Short-Term Load Forecasting (STLF) typically focuses on horizons ranging from one hour to a day ahead. These forecasts are critical for economic dispatch, unit commitment, load flow analysis, and real-time operation. Traditionally, utilities relied on statistical approaches such as linear regression, ARIMA, exponential smoothing, and Holt-Winters. These techniques can work well when patterns are simple, but they struggle with real-world load curves that are nonlinear, noisy, and influenced by many interacting variables.

Machine Learning models like Random Forest, Support Vector Regression, and XGBoost have shown strong results in several energy-related forecasting tasks. Their ability to capture nonlinear relationships makes them a natural fit for electricity load prediction. Likewise, Deep Learning approaches, especially recurrent neural networks such as LSTM and GRU, can learn temporal dependencies more effectively than traditional models.

The Baneshwor Feeder of the Nepal Electricity Authority serves a mixed group of consumers in the Baneshwor region. Its load pattern reflects residential lifestyles, commercial activity, seasonal tourism impacts, and local weather changes. Daily and weekly cycles are clearly visible, but there are also irregularities that simple models fail to capture. As power consumption continues to grow and diversify, the ability to forecast the feeder's short-term load accurately has become even more important. This creates a strong motivation to investigate how modern ML and DL models can improve forecasting performance for this specific feeder.

1.2 Problem Statement

The current forecasting practices for the Baneshwor Feeder rely heavily on manual estimation or basic statistical techniques. These methods do not fully capture the nonlinear and dynamic nature of the load profile, especially when multiple influencing factors, like temperature, humidity, rainfall, weekends, and special events come into play. As a result, prediction errors tend to increase during peak hours, sudden weather changes, and seasonal transitions.

Inaccurate short-term forecasts have several consequences. They can affect how generation is scheduled, leading to either unnecessary reserve margins or inadequate supply. They may increase operational costs and technical losses at the distribution level. In the worst cases, poor foresight during high-demand periods can create voltage drops, reliability concerns, or inefficient load-shedding decisions.

Despite the availability of historical load and weather data, there has not been a systematic study applying and comparing advanced machine learning and deep learning approaches specifically for the Baneshwor Feeder. The lack of a data-driven forecasting system means operators do not yet benefit from models that are capable of learning complex relationships within the data.

This thesis aims to address these gaps by building a complete forecasting framework using multiple ML and DL models, evaluating their performance, and identifying the most suitable approach for accurate short-term load prediction of the Baneshwor Feeder.

1.3 Objectives

To develop and evaluate machine learning and deep learning models for short-term electrical load forecasting of the Baneshwor Feeder to improve prediction accuracy and operational efficiency.

1. To collect and preprocess historical load data and relevant influencing factors such as weather variables and calendar effects for the Baneshwor Feeder.

2. To analyze consumption patterns and influencing factors (e.g., time, weather, festivals).

3. To implement various machine learning models, including Support Vector Regression (SVR), Random Forest (RF), and XGBoost, for load forecasting.

4. To design and train deep learning models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks to capture temporal dependencies in load data.

5. To evaluate and compare the performance of ML and DL models using standard error metrics (e.g., RMSE, MAPE, MAE, MSE, R-squared).

6. To compare and select the most effective forecasting model.

7. To recommend the most suitable forecasting model for operational use in the Baneshwor Feeder.

1.4 Scope

This study is geographically limited to the Baneshwor Feeder under the Nepal Electricity Authority. The temporal scope focuses on short-term load forecasting with a prediction horizon of up to 24 hours ahead, utilizing historical hourly load data as the primary foundation for model development. The dataset encompasses historical load data from the Baneshwor Feeder, complemented by weather data including temperature, humidity, and rainfall, along with calendar data distinguishing weekdays, weekends, and holidays.

From a technical perspective, the research implements several machine learning models including Support Vector Regression (SVR), Random Forest, and XGBoost, alongside deep learning architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. The performance of these models is rigorously evaluated using standard metrics including Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Mean Squared Error (MSE), and the coefficient of determination (R-squared). It should be noted that the accuracy of forecasts is inherently dependent on the quality and completeness of the historical data available. Additionally, this study does not extend to medium-term or long-term forecasting horizons, and the scope explicitly excludes renewable generation forecasting from its analysis.

1.5 Limitation

Despite the promising results obtained in this study, certain limitations were encountered, primarily related to data availability, model assumptions, and scope of analysis.

1. **Dependence on data quality:** Forecast accuracy is limited by the completeness and reliability of the historical load and weather data. Missing values, sensor errors, or inconsistent reporting can influence model performance.

2. **Model sensitivity to sudden changes:** Unexpected events such as outages, festivals, abrupt weather shifts, or abnormal consumption patterns are difficult for data-driven models to predict accurately.

3. **Deep learning computation constraints:** Training LSTM and GRU models requires more computational resources and time compared to ML models. Their performance may vary depending on the hardware used.

4. **Limited feature diversity:** Although weather and calendar data are included, other influential factors like economic activities, special events, or industrial load profiles are not part of the dataset.

5. **Generalization across feeders:** The models developed in this study are tailored specifically to the Baneshwor Feeder and may not generalize directly to other feeders without retraining or adaptation.

2. Literature Review

The literature review presents a comprehensive overview of existing research related to short-term electrical load forecasting using machine learning and deep learning techniques. It examines previously published studies to understand commonly used methodologies, datasets, and performance evaluation approaches in the domain of power system load forecasting. Reviewing prior work helps to identify current research trends, strengths, and limitations of existing models, while also highlighting gaps that motivate the need for this study. By situating the present research within the context of established knowledge, this chapter provides a foundation for model selection and methodological design adopted in this work.

2.1 Related work

There are many previous works done for electrical load forecasting from short-term electrical load forecasting, to medium-term and long-term. Most of the studies have done short-term load forecasting.

2.1.1 Machine Learning Approaches

Singla et al. (2019) employed Artificial Neural Networks for 24-hour short-term load forecasting, utilizing dew point temperature, dry bulb temperature, and humidity as input features. Their work demonstrated the effectiveness of ANN in capturing the relationship between weather variables and electrical load demand. Similarly, Desai et al. (2021) utilized the Prophet model from Meta to perform short-term load forecasting, incorporating time, temperature, humidity, and weather forecast data as features. The Prophet model's ability to handle seasonal patterns and missing data made it suitable for load forecasting applications.

Matrenin et al. (2022) conducted a study on medium-term load forecasting using ensemble machine learning models. They compared XGBoost and AdaBoost against traditional methods including SVR, decision trees, and Random Forest. Their results highlighted the superior performance of gradient boosting techniques for capturing complex load patterns. Aguilar Madrid & Antonio (2021) tested five machine learning models and found XGBoost to be the most accurate for predictions, using historical load data, weather information, and holiday indicators as input features. Their comprehensive evaluation demonstrated XGBoost's ability to handle diverse feature sets effectively.

Guo et al. (2021) analyzed three popular ML methods for load forecasting: Support Vector Machine, Random Forest, and LSTM. They proposed a fusion forecasting approach that combined outputs from all three models, demonstrating that ensemble methods could improve prediction accuracy beyond individual model performance. Saglam et al. (2024) performed a comparison between optimization methods (Particle Swarm Optimization, Dandelion Optimizer, Growth Optimizer) and machine learning models (SVR, ANN) for instantaneous peak electrical load forecasting. They found that ANN combined with Growth Optimizer outperformed other models and identified a strong positive correlation between GDP and peak load demand.

Jain & Gupta (2024) conducted a comprehensive evaluation of various machine learning algorithms for power load prediction, including Support Vector Machines, LSTM, ensemble classifiers, and Recurrent Neural Networks. Their study emphasized the importance of data preprocessing methods, feature selection strategies, and performance assessment metrics in achieving accurate forecasts. The research demonstrated that ensemble methods and deep learning approaches consistently outperformed traditional statistical models.

2.1.2 Deep Learning Architectures

Chapagain et al. (2021) explored time series regression along with machine learning and deep learning models for electricity demand forecasting in Kathmandu Valley. They found LSTM demonstrating outstanding performance in terms of MAPE and RMSE, using deterministic variables such as day type and temperature. Their work validated the effectiveness of recurrent architectures for capturing temporal dependencies in load data.

Acharya et al. (n.d.) performed short-term electrical load forecasting for the Gothatar feeder using six input features. They found that Recurrent Neural Networks outperformed baseline methods including Single Exponential Smoothing, Double Exponential Smoothing, and Holt-Winter's method. This study confirmed that RNNs could better model the non-linear and time-dependent characteristics of feeder-level load patterns.

Cordeiro-Costas et al. (2023) conducted a comprehensive comparison of load forecasting methods, including Random Forest, SVR, XGBoost, Multi-Layer Perceptron, and LSTM. They also explored Conv-1D models and found that LSTM achieved the lowest error rates across multiple evaluation metrics. Their research highlighted the trade-off between model complexity and forecasting accuracy in practical applications.

Dong et al. (2024) provided a comprehensive survey on deep learning-based short-term electricity load forecasting covering the past decade. They examined the entire forecasting process, including data preprocessing, feature extraction, deep learning modeling and optimization, and results evaluation. The survey identified CNN-LSTM hybrid architectures as widely adopted solutions due to exceptional performance in capturing both spatial and temporal features. Their analysis revealed that most recent studies focused on short-term horizons ranging from one hour to several days ahead.

2.1.3 Hybrid and Advanced Architectures

Wen et al. (2024) proposed a hybrid deep learning model combining Gated Recurrent Units and Temporal Convolutional Networks with an attention mechanism for short-term load forecasting. The GRU captured long-term dependencies in time series data, while TCN efficiently learned patterns and features. The attention mechanism automatically focused on input components most relevant to the prediction task, significantly enhancing model performance. Their approach demonstrated superior accuracy compared to standalone architectures.

Alhussein et al. (2020) developed a hybrid CNN-LSTM framework for short-term individual household load forecasting. The model used CNN layers for feature extraction from input data and LSTM layers for sequence learning. Evaluated on the Smart Grid Smart City dataset, the hybrid model achieved an average MAPE of 40.38%, outperforming standalone LSTM models that obtained 44.06% MAPE. This work demonstrated the effectiveness of combining convolutional and recurrent architectures for handling high volatility in household-level load data.

Hasanat et al. (2024) proposed a parallel multichannel network approach using 1D CNN and Bidirectional LSTM for load forecasting in smart grids. Unlike traditional stacked CNN-LSTM architectures that use convolutions as preprocessing steps, their model independently processed spatial and temporal characteristics through parallel channels. The research addressed the issue of temporal feature neglect in existing models and incorporated cyclic features through trigonometric transformations, achieving superior accuracy on diverse building types.

2.1.4 Transformer-Based Models

Chan & Yeo (2024) proposed a sparse transformer-based approach for electricity load forecasting that addressed the computational complexity limitations of standard transformer architectures. Their model applied sparse attention mechanisms to capture temporal dependencies more efficiently, achieving comparable accuracy to RNN-based state-of-the-art methods while being up to 5 times faster during inference. The model was enhanced to support multivariate inputs including weather data, demonstrating versatility in forecasting loads from individual households to city levels.

Zhang et al. (2022) developed a Time Augmented Transformer model for short-term electrical load forecasting, incorporating temporal features and self-attention mechanisms to capture complex dynamic nonlinear sequence dependencies. Their experimental results showed that multivariate inputs including weather and calendar features produced significantly better predictions than univariate approaches. The attention mechanism's capacity to capture complex dynamical patterns in multivariate data contributed to improved forecasting accuracy.

Lu & Chen (2024) proposed a multivariate data slicing transformer neural network for load forecasting in power systems with high-penetration renewables. The transformer model excelled in capturing spatiotemporal relationships by modeling global correlations through self-attention mechanisms. Their approach demonstrated superior performance in handling the intermittency and volatility characteristics brought by renewable energy integration, outperforming traditional statistical models and conventional machine learning methods.

2.1.5 Comparative Studies and Ensemble Methods

Banik & Biswas (2024) developed an enhanced stacked ensemble model combining Random Forest and XGBoost for renewable power and load forecasting. The Random Forest model first forecasted the target variable, followed by XGBoost improving predictions through combination of RF outputs. A meta-model using logistic regression then learned the optimal combination, achieving 99% accuracy on R^2 evaluation metrics for both short-term and long-term predictions in Agartala City dataset.

Kwon et al. (2020) conducted extensive research on learning models combined with data clustering and dimensionality reduction for short-term electricity load forecasting. They adapted k-means clustering for data grouping and utilized kernel PCA, UMAP, and t-SNE for dimensionality reduction. Applied to neural network-based models on large-scale electricity usage data from 4,710 households, their approach demonstrated improved forecasting performance through effective data preprocessing and feature engineering.

Nabavi et al. (2024) combined Discrete Wavelet Transform with LSTM to improve electricity load forecasting accuracy. The DWT decomposed load series into multiple frequency components, allowing LSTM to learn from denoised and structured representations. Their research demonstrated that preprocessing techniques significantly enhanced deep learning model performance, particularly for datasets with high noise levels and irregular patterns.

3. Methodology

This chapter describes the overall research methodology adopted to achieve the objectives of the thesis. It outlines the research design, data acquisition process, preprocessing techniques, and modeling approaches used for short-term electrical load forecasting. The methodology is structured to ensure a systematic analysis, with each step directly linked to the research objectives. A flowchart is used to summarize the overall framework, followed by a detailed explanation of the selected machine learning and deep learning models and the evaluation techniques employed in this study.

3.1 Overall Workflow

The forecasting framework used in this thesis follows a clear and systematic workflow. Each stage builds on the previous one to ensure that the final model is both reliable and reproducible. The complete process can be broken into six major steps:

1. **Data Acquisition:** Hourly historical load data from the Baneshwor Feeder is collected alongside weather variables (temperature, humidity, rainfall) and calendar information (weekday/weekend, holidays). These serve as the core inputs to the forecasting models.

2. **Data Preprocessing:** The raw dataset often contains missing readings, outliers, and format inconsistencies. This stage involves cleaning the data, handling missing values, treating outliers, converting timestamps, and engineering new time-based and cyclical features. This step ensures that the dataset is suitable for model development.

3. **Exploratory Data Analysis (EDA):** The cleaned dataset is examined to understand load trends, seasonal patterns, hourly variations, and correlations with weather variables. EDA helps identify which features influence load demand the most and guides feature selection.

4. **Model Development:** Both machine learning models (SVR, Random Forest, XGBoost) and deep learning models (LSTM, GRU) are designed. Input features are standardized and structured depending on the model type—tabular matrices for ML models and sequential inputs for DL models.

5. **Model Training and Validation:** Models are trained using training data and validated using walk-forward validation or hold-out testing. Hyperparameters are op-

timized (GridSearchCV for ML models and iterative tuning for DL models). This ensures that the models generalize well and avoid overfitting.

6. **Performance Evaluation and Comparison:** All models are evaluated using RMSE, MAE, MAPE, MSE, and R^2. Their forecasting accuracy, stability, and computational efficiency are compared. The best-performing model is then recommended for operational forecasting in the Baneshwor Feeder.

This workflow provides a complete pipeline from raw data to final recommendation and supports both machine learning and deep learning approaches. The overall diagram of the methodology is shown in Figure 3.1.

3.2 Data Acquisition

The first step of our methodology is data acquisition. Hourly historical load data from Baneshwor Feeder will be collected. Weather data such as temperature, humidity, and rainfall data will be collected from the Department of Hydrology and Meteorology, Nepal. And finally, the information on weekdays, weekends are obtained from official government calendars. The study relies on two primary datasets:

1. Hourly electrical load data for the Baneshwor Feeder
2. Hourly weather data (temperature, humidity, and solar radiation)

Since both datasets were obtained as raw Excel files with irregular formats, inconsistent timestamps, missing entries, and multiple sheets per day/month, a multi-stage acquisition and structuring pipeline was developed. The sources of both datasets are listed below.

3.2.1 Data Sources

a) **Electrical Load Data:** The electrical load data used in this study was obtained from the Baneshwor Substation, which operates under the Nepal Electricity Authority (NEA). Hourly feeder load readings were collected from archived operational log sheets maintained by the substation for the years 2079 to 2082 BS. These records provided raw POWER (MW) measurements for the Baneshwor Feeder, along with associated timestamp information. Since the data originated from manually recorded and distributed Excel files, several preprocessing steps—such as header correction, timestamp standardization, and quality checks—were required before the dataset could be used for modeling. This substation-provided dataset forms the core of the forecasting analysis, representing real operational feeder behavior across multiple years.

b) **Weather Data:** Weather data was sourced from Nepal's Department of Hydrology and Meteorology (DHM), the official governmental agency responsible for climate and atmospheric measurements. The dataset included hourly records of air temperature, relative humidity, and global solar radiation for the corresponding study period. These variables were essential for capturing the environmental conditions influencing electricity consumption patterns. The DHM dataset required timestamp alignment, interpolation for missing values, and smoothing of extreme readings to ensure compatibility with the load dataset. Once cleaned and synchronized, the weather data served as an important set of exogenous features for both machine learning and deep learning models.

Because the raw files came in varying formats—different months, unpredictable sheet names, Bikram Sambat (BS) dates, mixed day formats, half-hour readings, inconsistent header rows, and multiple sheets per month—a custom data acquisition pipeline was required.

3.2.2 Load Data Acquisition and Structuring Process

The original NEA-provided Excel files were not uniform. Each month contained multiple workbooks, each workbook contained several sheets, and each sheet sometimes mixed headers, irrelevant rows, or inconsistent timestamp formats. To convert all raw data into a single unified structure, the following pipeline was implemented:

a) **Month-wise Sheet Extraction:** The script automatically scanned each monthly folder, such as those inside dataset/2082MW, and identified all sheets whose names contained "11KV" or variations of it. From there, it matched the sheet names to the correct year–month to ensure proper alignment during extraction. Only rows with timestamps ending in HH:00 were selected, and half-hour readings like 7:30 were intentionally ignored. For every sheet, the script compiled a list of valid hourly entries and then stored the extracted results month-wise inside the corresponding extractedDataset/YYYYMM folders. This step produced a clean per-month dataset, but timestamps were still in Nepali calendar (BS) and formats varied.

b) **Converting BS Dates, Normalizing Hours, and Structuring Daily Sheets:** The process began by extracting the BS date encoded inside each sheet name, such as 2079.02.15, and converting it to the AD calendar using Nepali date conversion libraries. Each sheet was read without assuming a fixed header position, allowing the script to dynamically detect both the Time column and the appropriate POWER (MW) value column. Every day was normalized to 24 hourly records by indexing hours from 1 to 24, and any missing hours were filled using linear interpolation. Clean timestamps were then constructed in the formats YYYY-MM-DD HH:00 and YYYY-MM-DD 24:00 for the final hour. Finally, one clean sheet was generated per day in the AD calendar, ensuring that each day contained exactly twenty-four readings aligned to a consistent timestamp system.

c) **Combining All Structured Monthly Files:** The script scanned the four yearly folders from 2079 to 2082, and for each folder it opened every Excel file, extracted the valid Time and POWER (MW) columns, removed any remaining header fragments, and concatenated all sheets in that folder into a single file. After processing all four years, the resulting files were merged into one master dataset. At this stage, the study had a fully consolidated file containing all hourly POWER (MW) readings for the entire study period.

3.2.3 Weather Data Acquisition and Structuring

Weather data was also provided in raw format with mixed timestamps. Two scripts were developed to clean and align it with the load data.

a) **Extracting and Cleaning Raw Weather File:** The script located the correct columns for time, temperature, humidity, and solar radiation, then removed any unusable rows. All timestamps were parsed into a consistent datetime format, after which the weather data was filtered to match the exact date range of the load dataset. The timestamps were then formatted as YYYY-MM-DD HH:MM. This stage produced a clean hourly weather dataset.

b) **Structuring Weather Timestamp Alignment:** Timestamps were shifted so that values such as "HH:45" were aligned to the next hour at "HH+1:00," and all "24:00" rollover cases were handled correctly. Missing or zero weather values were replaced using nearest-neighbor averages, while NaN solar radiation entries were set to zero. These steps produced the final clean weather file and ensured that all weather variables followed the exact hourly structure required for forecasting.

3.2.4 Final Merging of Load and Weather Data

The process involved parsing all load timestamps, including proper handling of "24:00," and parsing the weather timestamps in a consistent format. Weather rows were then aligned precisely to the corresponding load timestamps, and any gaps in the weather variables were filled using linear interpolation. Finally, both datasets were merged into a single unified file. This produced the final dataset containing:

1. Time
2. Power (MW)
3. Air Temperature
4. Global Solar Radiation
5. Relative Humidity

This dataset is the core input for all ML/DL models used in the thesis.

4. Experimental Setup (if any)

In this section, you describe how the experiment was done and summarize how the data was taken. One typically describes the instruments and detectors that were used. Describe the procedure that was followed to collect the data etc.

5. Results & Discussion

This section should present the findings of the study in logical sequences in line with the specific objectives. The presentation of data and facts should be explained regarding plausibility and compared with data from similar studies. The causal factors behind the findings should be discussed about other variables under consideration in the study based on Focused Group Discussion (FGD), Key Informant Interview (KII), questionnaire survey, modeling, observation, measurement, or literature reviews.

6. Conclusions & Recommendations

The conclusion is an integration of various issues covered in the body of the thesis. The conclusion includes noting any implications resulting from the discussion and making policy recommendations and the need for further research. Hence, the conclusion should be a logical ending to what has been previously discussed. It must pull together all parts of the argument and refer the reader back to the focus you have outlined in your introduction and to the central topic. Never present any new information in this section. Thus, the conclusion and recommendation of the study must be limited within the scope of the research.

7. Limitations and Future enhancement

This chapter should contain the major limitations of the project and the further enhancement of the project/research shortly with a different but related approach. **Referencing checking here**[1]

References

[1] Santosh Giri and Basanta Joshi. Transfer learning based image visualization using cnn. *International Journal of Artificial Intelligence & Applications*, 10(4):47–55, 2019.

Appendices

This page contains a data sheet, coding, procedure, photograph, questionnaire, and other essential documents. This page should be started from an odd page and APPENDIX numbering should be A, B, C, etc.