Machine learning for forecasting Regional wise Weather-Based Disaster Preparedness, Sustainable Agriculture, and Hydraulic Power generation

C.P.Weerasinghe
IT20148836

B.Sc. (Hons) Degree in Information Technology
Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

September 2023

Weather Prediction and Optimal Power Generation for Hydraulic Turbines: A Machine Learning Approach

C.P.Weerasinghe

IT20148836

B.Sc. (Hons) Degree in Information Technology

Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

Declaration

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

IT20148836	**C.P.Weerasinghe**	

Date: 10/06/2023

Abstract

Hydropower, as one of the oldest forms of renewable energy, has historically played an important role in powering human civilizations. Today, it remains an important component of the global renewable energy landscape, providing sustainability and dependability. The importance of optimizing hydropower generation and efficiently managing water resources in the face of changing weather conditions is emphasized in this report. Water availability challenges are inextricably linked to weather conditions, which can vary dramatically across seasons and regions. Water resource ebbs and flows have a direct impact on energy production, causing crises during dry spells and abundance during rainy seasons. To address these fluctuations and provide people with consistent energy access, we are investigating the feasibility of increasing hydropower production during rainy seasons as a viable solution.

Traditional energy generation methods, such as coal thermal power, are environmentally unsustainable in the long run. As a result, increasing hydraulic power capacity during rainy seasons becomes critical in addressing energy crises and reducing reliance on nonrenewable sources. This report describes a novel approach that uses machine learning to predict energy production in various weather conditions. Individuals and organizations can plan and operate more resiliently, reducing the risk of energy shortages, by forecasting energy consumption patterns tailored to specific climates and months. This predictive capability represents a promising avenue for mitigating energy crises and promoting sustainable energy practices.

Keywords: Hydropower, Optimal power generation, Machine Learning, Energy production predictions, Time Series

Acknowledgement

I would like to express my heartfelt gratitude to my supervisor, Ms. Vindhya Kalapuge, for her invaluable guidance and motivation, which were critical in the success of this research. I am also grateful for the unwavering support of the Department of Information Technology at the Sri Lanka Institute of Information Technology, as well as the dedicated CDAP lecturers and staff. Their advice and assistance were extremely helpful in this endeavor.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

1. ANNs - Artificial Neural Networks
2. SVMs - Support Vector Machines
3. RFs - Random Forests
4. RNN - Recurrent Neural Network
5. SVR - Support Vector Regression
6. LSTM - Long Short-Term Memory
7. RMSE - Root Mean Squared Error
8. MSE - Mean Squared Error
9. R2 Score - R-squared Score
10. GPR - Gaussian Process Regression
11. Auto ARIMA - Automated AutoRegressive Integrated Moving Average
12. MAE - Mean Absolute Error
13. AIC - Akaike Information Criterion

List of Appendices

1.0 INTRODUCTION

Hydropower, which is derived from the kinetic energy of flowing water, has a long history as one of humanity's first sources of mechanical power. Water wheels and mills, which date back to ancient civilizations, paved the way for agricultural and industrial revolutions, fundamentally shaping our societies. Today, hydropower is still relevant, but in a very different context: as a key component of renewable energy generation. Hydropower is a modern-day symbol of sustainability and resilience in the face of climate change. Its significance is inextricably linked to the pressing need to optimize energy generation while efficiently managing our planet's most precious resource - water. The significance of optimizing hydropower generation cannot be overstated, as it represents a clean and renewable energy source that can significantly alleviate energy crises, particularly during changing weather patterns.

Weather conditions have a significant impact on the availability of water resources, resulting in a dynamic energy landscape. In times of heavy rainfall, hydropower systems provide increased energy potential, bolstering our energy reserves. During dry seasons, however, we face the harsh reality of energy scarcity. It is precisely at this juncture that the art of predictive modeling becomes indispensable. When we try to meet diverse energy demands across different weather patterns, we frequently find ourselves in crisis situations. Understanding and harnessing the full potential of hydropower during rainy seasons can provide a lifeline during dry spells, avoiding the need to rely on less sustainable energy sources such as coal thermal power. Prolonged reliance on such alternatives is unsustainable and detrimental to our environment.

The ability to accurately predict energy supply under varying weather conditions is crucial to averting these energy crises. Energy consumption patterns differ not only by season, but also by cultural and climatic events. Consider the increased energy demand during festive seasons such as Christmas or cultural celebrations such as Vesak season. Precise energy supply forecasts offer the promise of continuous energy supply, fundamentally changing people's lives. Hydropower, with its adaptability and environmental friendliness, plays a critical role in this transformation. We can pave the way for a future where energy crises are relegated to the annals of history by comprehensively understanding its interplay with alternative energy sources and providing accurate future energy supply forecasts. This report delves into the art of forecasting, examining how it can protect us from the looming threat of energy crises and ensure sustainable access to energy regardless of weather conditions or shifting demand.

Hydropower has been a significant source of renewable energy for centuries, with the first recorded use of water wheels dating back to Ancient Greece around 200 B.C. [1]. In the modern era, hydropower has become an essential contributor to the global energy mix, providing approximately 16% of the world's electricity in 2019. With the global focus on reducing carbon emissions and transitioning towards more sustainable energy sources, the importance of hydropower as a renewable and reliable energy source is only expected to grow [2]. Despite the advantages of hydropower, it is not without its challenges. One of the primary concerns in hydropower generation is the reliance on water availability, which is heavily influenced by weather conditions such as precipitation, temperature, and evaporation rates [3]. Accurate prediction of these variables is essential for efficient power generation using hydraulic turbines, as it enables better planning and management of water resources [3]. In addition, understanding the patterns of power consumption in each area can help operators optimize the use of available resources and reduce the need for power cuts or reliance on non-renewable energy sources [4].

Over the past few decades, significant advancements have been made in the field of weather prediction, primarily due to the development of more sophisticated numerical weather prediction models [5] and the increasing availability of high-resolution satellite data [6]. In parallel, the growth of machine learning techniques has also contributed to improvements in weather forecasting, with studies showing that machine learning models can outperform traditional statistical methods in predicting variables such as precipitation [7] and temperature [8]. Several machine learning techniques have been successfully applied to predict power generation from renewable sources, including artificial neural networks (ANNs) [9], support vector machines (SVMs) [10], and ensemble methods like random forests (RFs) [11]. Many of these studies have focused on solar and wind power generation, but there has also been growing interest in applying these techniques to hydropower prediction [12]. In recent years, researchers have started to explore the potential of machine learning in predicting power consumption patterns as well. For instance, [13] employed a clustering algorithm to group households with similar consumption patterns, while [14] used a deep learning model to forecast the electricity load of a large city. These studies demonstrate the potential of machine learning techniques in understanding and predicting the complex relationships between weather conditions, power generation, and power consumption.

Despite the promising results of these studies, there remains a need for an integrated approach that combines weather prediction, power generation optimization, and power consumption forecasting to improve the efficiency of hydraulic turbines in various weather conditions. This research aims to address this gap by developing a machine learning-based framework that can predict the optimal times for power generation using hydraulic turbines based on weather conditions and power consumption patterns in a specific area. The development of a machine learning-based approach for predicting the optimal times for generating power using hydraulic turbines has the potential to significantly contribute to the efficient management of hydropower resources. By integrating weather prediction and power consumption forecasting models, this research can help minimize power cuts and reduce the need for non-renewable energy sources, ultimately benefiting both consumers and the environment.

1.1 Background and Literature

Power Generation Prediction

Accurate prediction of power generation is essential for the efficient operation and management of hydropower facilities. Various machine learning techniques have been employed to predict power generation from renewable sources, including wind, solar, and hydropower. [8] used an artificial neural network (ANN) to predict the output of a photovoltaic system in Algeria, demonstrating the effectiveness of machine learning in modeling complex relationships between weather variables and power generation. In the context of hydropower, [22] conducted a comprehensive review of artificial intelligence-based prediction models and identified several successful applications of ANNs, SVMs, and other machine-learning techniques for hydropower prediction. [23] developed a recurrent neural network (RNN) model to predict the hourly water inflow in a hydropower reservoir, demonstrating improved accuracy compared to traditional time series models. More recently, [23] applied a support vector regression (SVR) model to predict hydropower generation in a Turkish dam and observed better performance than other traditional and machine learning models.

Power Consumption Prediction

Power consumption prediction is critical for efficient energy management and reducing reliance on nonrenewable energy. Machine learning is widely used to forecast consumption patterns at all scales, from homes to cities. [24] Employed a clustering algorithm to group households based on their electricity consumption patterns, demonstrating the potential of unsupervised learning techniques in understanding consumer behavior. In a large-scale study, [23] used a deep learning model, specifically a long short-term memory (LSTM) network, to forecast the hourly electricity demand of a city. The LSTM model achieved better accuracy than traditional time series models. Several studies have focused on the relationship between weather conditions and power consumption. Hong et al. developed a machine learning-based model to predict residential heating energy consumption using weather data, building characteristics, and historical energy consumption data. Their model outperformed traditional linear regression models. Similarly, [25] used an artificial neural network (ANN) to predict monthly electricity demand in Saudi Arabia based on weather variables and socioeconomic.

1.1.1 Centrality Measures

Centrality measures are important tools in network analysis because they help us identify the most influential nodes in a network. Centrality measures are critical in identifying key elements within the complex web of weather conditions, power generation, and consumption patterns in the context of hydropower optimization and energy supply forecasting. These metrics assist us in identifying the nodes that have the most influence over the system's efficiency and resilience, whether they represent geographic locations, specific weather variables, or power consumption hubs. By analyzing centrality, we can learn which areas are most important for optimal power generation planning, where weather conditions have the most influence, and which power consumption patterns drive energy demand. This data is invaluable for developing a machine learning-based framework capable of accurately predicting optimal power generation times, reducing power outages and reliance on nonrenewable energy sources. In essence, centrality measures serve as a compass, guiding us toward a more efficient and sustainable future in hydropower generation and energy supply forecasting.

1.1.1.1 Degree Centrality

Degree centrality is a fundamental concept in network theory and analysis. It quantifies the importance or centrality of a node within a network by counting the number of connections it has to other nodes. In the context of your research problem, degree centrality can be applied to assess the significance of various energy sources within the integrated framework. By analyzing the connections and interactions between different energy generation methods, including hydraulic power, thermal coal power, and renewable sources, we can identify which sources are most central in meeting energy demands under varying weather conditions. This analysis can guide decisions on resource allocation and help optimize energy generation, ultimately reducing energy crises and environmental impacts.

1.1.1.2 Closeness Centrality

Closeness centrality is a pivotal concept in network analysis, measuring how quickly a node can interact with others within a network. In the context of our integrated hydropower optimization framework, it plays a crucial role in determining the most efficient times to generate power. By evaluating the closeness centrality of weather prediction, power generation, and power consumption models, we can identify key nodes in the network, facilitating timely decision-making to mitigate energy crises, reduce environmental impact, and ensure the seamless integration of diverse energy sources.

1.1.1.3 Betweenness Centrality

In the realm of optimizing hydropower generation, Betweenness Centrality emerges as a vital concept. It quantifies the criticality of specific nodes within a network, helping identify key junctures in the integration of weather prediction, power generation optimization, and power consumption forecasting. By pinpointing the nodes with the highest Betweenness Centrality, we can strategically allocate resources and interventions to maximize the efficiency of hydropower systems and minimize energy crises, particularly in comparison to non-renewable energy sources like coal thermal power.

1.1.1.4 Eigenvector Centrality

The absence of an integrated approach that harmonizes weather forecasting, hydraulic power generation prediction, and power consumption modeling to optimize hydraulic turbine operations across varying weather conditions is the research problem addressed in this study. This issue is of paramount importance, given the historical significance of hydropower and its current role as a renewable energy source. Water resource management is difficult in the face of changing weather patterns, and integrating these models can lead to advanced predictions, minimizing energy crises while incorporating multiple energy sources. The proposed framework promises improved resource management as well as fewer power outages and a smaller environmental footprint for consumers.

1.2 Research Gap

While the literature review demonstrated the success of machine learning techniques in the domains of weather prediction, power generation prediction, and power consumption prediction, there is a significant research gap in the development and implementation of an integrated approach that combines these components to optimize power generation using hydraulic turbines under various weather conditions. The majority of studies in the literature have concentrated on discrete aspects of the problem, such as weather forecasting or power consumption forecasting, without taking into account the complex interdependencies between weather conditions, power generation, and power consumption patterns. As a result, these studies may not fully capture the potential benefits of a comprehensive approach that addresses the optimization problem.

Furthermore, previous research that proposed integrated approaches for hydropower optimization did not specifically address hydraulic turbines and their unique operational constraints. As a result, there is a need for a customized solution that takes into account the unique characteristics of hydraulic turbine-based power generation systems. This research gap provides an opportunity to develop a machine learning-based framework that can predict the optimal times for power generation using hydraulic turbines based on weather conditions and power consumption patterns in a specific area. This research can contribute to the efficient management of hydropower resources by integrating weather prediction, power generation prediction, and power consumption prediction models, thereby minimizing power outages and reducing the need for nonrenewable energy sources. Finally, by improving the efficiency and sustainability of hydropower generation, this research has the potential to benefit both consumers and the environment.

2.0 RESEARCH PROBLEM

"How can an integrated approach harmonize hydropower with alternative energy sources across varying weather conditions, resulting in an effective dynamic energy ecosystem able to reduce reliance on nonrenewable energy and maximize resource utilization?"

The purpose of this research problem is to facilitate the transition from conventional energy reliance to the harmonious coexistence of hydropower and alternative sources of energy. In addition to ensuring continuous and reliable energy flow to consumers, it aims to minimize the environmental impact of energy supply under diverse weather conditions. The global energy landscape is at a crossroads, grappling with the dual challenges of climate change and energy security. Traditional energy sources, such as coal thermal power, are unsustainable and environmentally harmful. Hydropower, on the other hand, is a renewable and cleaner alternative, but its efficiency is intricately linked to weather conditions.

The study investigates how an integrated energy approach can balance energy supply during variable weather conditions to address this. Hydropower can thrive during rainy seasons, but it is critical to harness this abundance while also optimizing the integration of alternative energy sources, such as solar and wind power, to ensure uninterrupted energy access. Investigate methods to maximize hydropower generation during rainy seasons, reducing the need for nonrenewable energy sources and ensuring consistent power supply even during dry periods. This includes researching advanced hydraulic turbine designs as well as water management strategies. Investigate how hydropower can be seamlessly integrated with alternative energy sources such as solar and wind to create a hybrid energy system that can adapt to changing weather conditions. This involves developing predictive models for each energy source and optimizing their coexistence.

The successful implementation of this integrated approach has the potential to improve energy security by reducing power outages and eliminating energy crises during dry seasons. Simultaneously, it will contribute to environmental sustainability by reducing carbon emissions through reduced reliance on non-renewable energy sources.

As a result of this research problem, the ebb and flow of nature can be seamlessly incorporated into energy production, thereby minimizing disruptions and safeguarding the environment.

3.0 OBJECTIVES

3.1 Main Objective

The main objective of this research is to develop an integrated machine learning-based framework that combines weather prediction, power generation prediction, and power consumption prediction models to optimize the operation of hydraulic turbines under various weather conditions in a specific area.

3.2 Specific Objectives

The following are the specific goals of this study,

1. Develop accurate machine learning models for predicting weather conditions, such as precipitation, temperature, and evaporation rates, that influence water availability for hydropower generation in the target area.
a. Collect historical weather data for the target area, including precipitation, temperature, and evaporation rates.
b. Preprocess and analyze the data to identify relevant patterns and relationships.
c. Evaluate and select appropriate machine learning algorithms for weather prediction.
d. Train, validate, and fine-tune the selected algorithms using the historical weather data.
e. Assess the accuracy and reliability of the developed weather prediction models.

2. Create machine learning models that predict power generation from hydraulic turbines based on the predicted weather conditions and the specific characteristics of the hydropower system.
a. Collect data on hydraulic turbine operations, including power generation, efficiency, and operational constraints.
b. Preprocess and analyze the data to identify relevant patterns and relationships.
c. Evaluate and select appropriate machine learning algorithms for power generation prediction.
d. Train, validate, and fine-tune the selected algorithms using the collected data.
e. Assess the accuracy and reliability of the developed power generation prediction models.

3. Develop machine learning models to forecast power consumption patterns in the target area, taking into account the complex relationships between weather conditions, consumer behavior, and other relevant factors.
a. Collect historical power consumption data for the target area, along with relevant demographic, socioeconomic, and behavioral factors.
b. Preprocess and analyze the data to identify relevant patterns and relationships.
c. Evaluate and select appropriate machine learning algorithms for power consumption prediction.
d. Train, validate, and fine-tune the selected algorithms using the collected data.
e. Assess the accuracy and reliability of the developed power consumption prediction models.

4. Integrate the weather prediction, power generation prediction, and power consumption prediction models to determine the optimal times for generating power using hydraulic turbines.

a. Develop a decision-making algorithm that considers the predictions from the three models to optimize the operation of hydraulic turbines.

b. Validate and fine-tune the decision-making algorithm using historical data and performance metrics.

c. Evaluate the effectiveness of the integrated approach in optimizing hydraulic turbine operations under various weather conditions.

5. Assess the potential impact of the proposed framework on the efficient management of hydropower resources, including minimizing power cuts and reducing reliance on non-renewable energy sources.

a. Conduct a case study to compare the performance of the proposed framework with traditional power management methods in the target area.

b. Analyze the potential environmental and economic benefits of implementing the proposed framework.

c. Identify possible limitations and areas for future research and improvement in the developed framework.

4.0 METHODOLOGY

4.1 Requirement Gathering

The methodology of this research consists of several stages, including data collection and preprocessing, model development, model integration, and evaluation. The following sections provide a detailed description of each stage of the methodology.

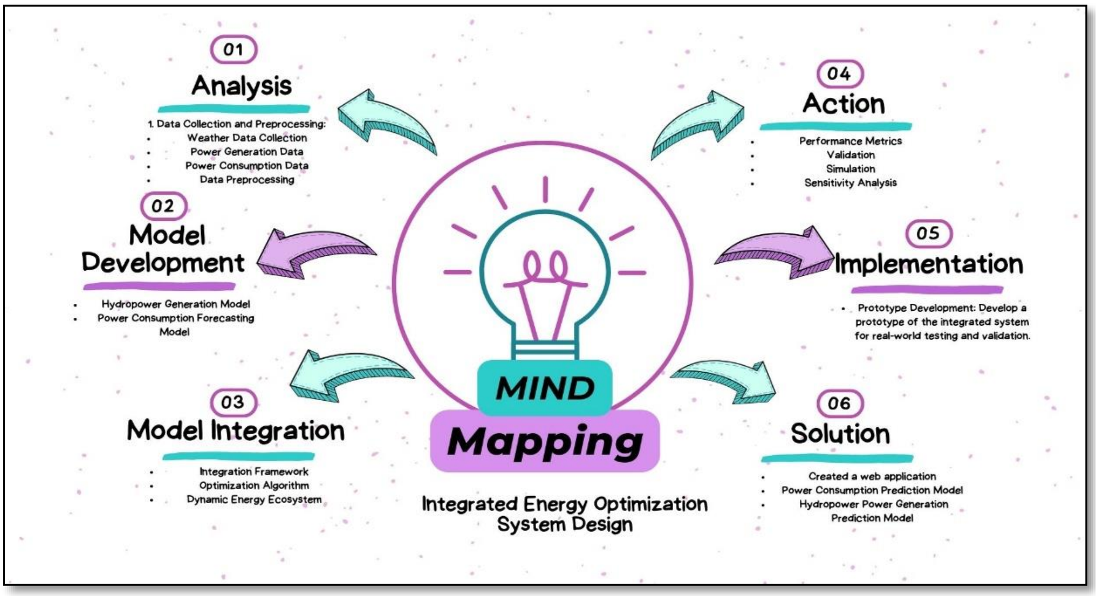


Figure 1.1: System Requirements Mapping Diagram.

4.1.1 past Research Analysis

Recent research in hydropower and energy forecasting has established a solid foundation for understanding the complex relationship between weather conditions, power generation, and consumption. While advances in numerical weather prediction models and the use of machine learning techniques have significantly improved weather forecasting accuracy, they have primarily been applied to solar and wind energy prediction. Only a few studies have looked into machine learning's potential in hydropower prediction and power consumption pattern analysis. Existing research emphasizes the importance of an integrated approach that combines weather prediction, power generation optimization, and consumption forecasting. This study aims to close the gap by creating a specialized machine learning-based framework that promises optimized hydropower generation and a more sustainable energy future.

Previous research analysis was primarily accomplished by reading research publications focusing on key areas such as short term load prediction, time series analysis, and machine learning models.

The primary emphasis was on identifying the methodology used, experiments conducted, and overall research findings regarding load forecasting and resource utilization prediction.

4.1.2 Identifying Existing Systems

Existing hydropower optimization systems primarily focus on isolated aspects such as weather forecasting, power generation forecasting, or power consumption analysis. While numerical weather prediction models and the use of machine learning techniques to improve weather forecasting accuracy have made significant advances, these systems frequently lack integration with hydropower generation optimization. Some studies have investigated machine learning for predicting hydropower generation or power consumption patterns separately, but these elements have not been effectively combined into a cohesive framework. This study aims to close these gaps by developing a comprehensive, machine learning-based approach that combines weather prediction, power generation optimization, and power consumption forecasting to improve the efficiency and sustainability of hydraulic turbine-based power generation.

4.2 Feasibility Study

4.2.1 Technical Feasibility

The technical feasibility of this research endeavor is supported by the convergence of several key factors. For starters, advances in machine learning and numerical weather prediction models have reached a level of sophistication that allows weather forecasts to be integrated with power generation optimization and consumption forecasting. Furthermore, the abundance of high-resolution satellite data and the demonstrated success of machine learning in predicting weather variables like precipitation and temperature provide a solid foundation for this endeavor. Furthermore, the successful application of machine learning techniques in the renewable energy sector, particularly in predicting power generation from sources such as solar and wind, demonstrates the adaptability and viability of these methods. Given these advancements, as well as the emphasis on hydraulic turbines, a mature and well-established technology, the technical feasibility of developing a machine learning-based framework to predict optimal hydropower generation times is clear.

4.2.2 Schedule Feasibility

Implementing an integrated machine learning-based framework for optimizing hydropower generation presents a multifaceted challenge that includes data acquisition, model development, and real-world deployment. The timeline for this research project will encompass several critical phases, including data collection and preprocessing, model training and validation, and integration into existing hydropower infrastructure. Given the complexity of these tasks and the need for thorough testing and validation, a realistic schedule will need to be established. Delays could arise from data availability, model refinement, and unexpected technical issues. Therefore, meticulous project management and a flexible timeline are essential to ensure the successful development and deployment of the predictive framework within the allocated timeframe. The project schedule will be a vital aspect of ensuring the feasibility of this research and its potential impact on efficient hydropower resource management.

4.2.3 Economic Feasibility

A machine learning-based framework for optimizing hydropower generation through integrated weather prediction, power generation optimization, and power consumption forecasting is essential. Long-term benefits far outweigh initial development and implementation costs. In periods of hydropower shortage, the framework can minimize reliance on costly non-renewable energy sources by enhancing the efficiency of hydraulic turbines under varying weather conditions. The result can be substantial economic gains, increased energy security, and reduced environmental impact. Sustainable energy solutions are economically viable when they are aligned with global efforts to mitigate climate change and transition to a greener energy landscape.

4.3 Requirement Analysis

The requirement analysis for this research includes a thorough investigation into the multifaceted components of an integrated machine learning-based framework for optimizing hydropower generation using hydraulic turbines under varying weather conditions. This analysis entails gathering and preprocessing historical data on weather conditions, power generation, and power consumption patterns. It also entails the evaluation and selection of appropriate machine learning algorithms for predicting weather variables, power generation outcomes, and power consumption behavior.

Furthermore, the analysis includes the development of decision-making algorithms that seamlessly integrate these predictive models to determine the best times for hydropower generation. This includes validation and fine-tuning processes to ensure the framework's accuracy and dependability. In addition, a case study is carried out to assess the potential impact of the proposed framework on the efficient management of hydropower resources, with an emphasis on its ability to minimize power outages and reduce reliance on non-renewable energy sources. Throughout this analysis, the identification of limitations and areas for future research and improvement remains a crucial aspect of the research endeavor.

4.4 System Analysis

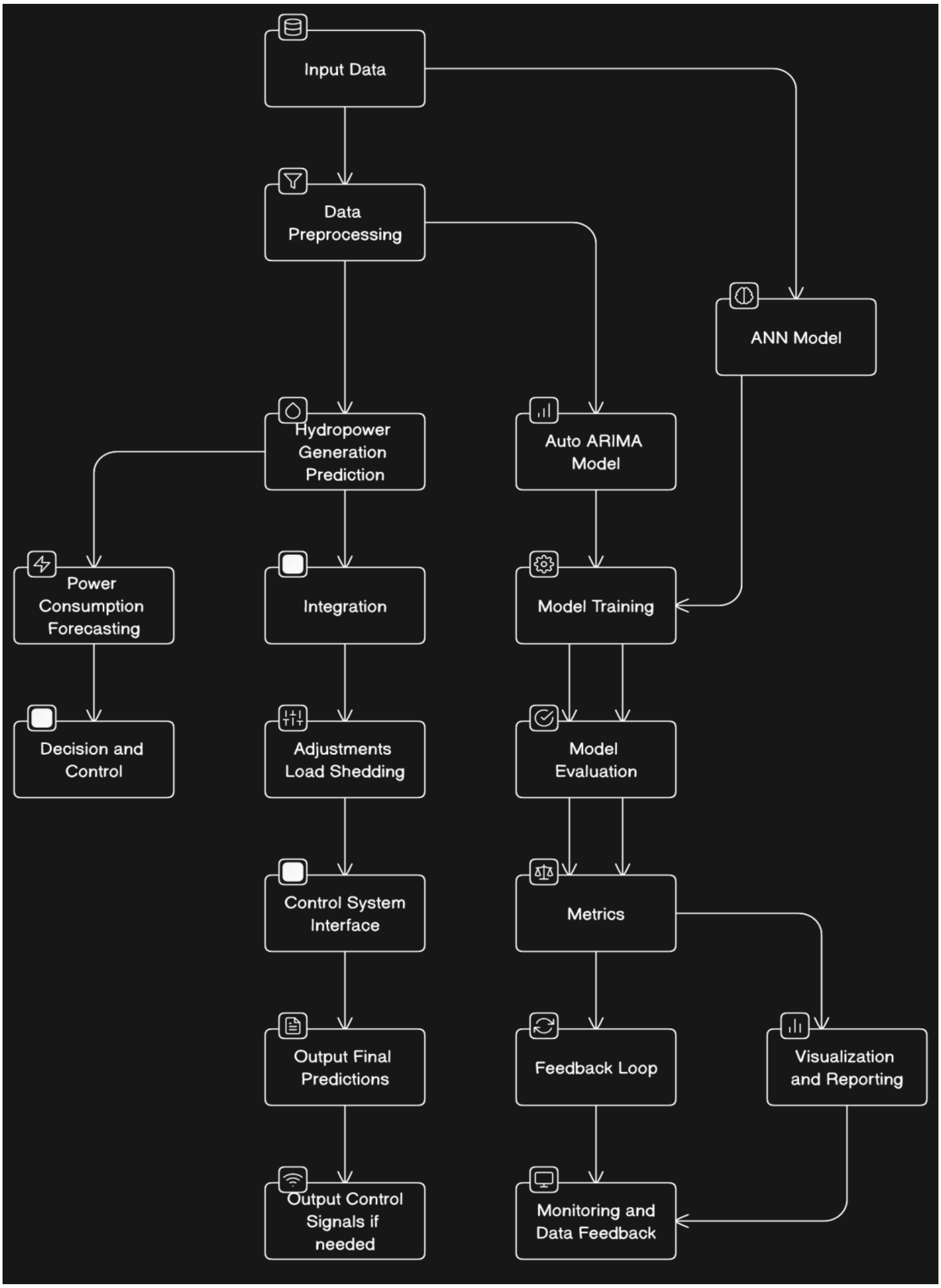


Figure 4.4System diagram

4.5 System Development and Implementation

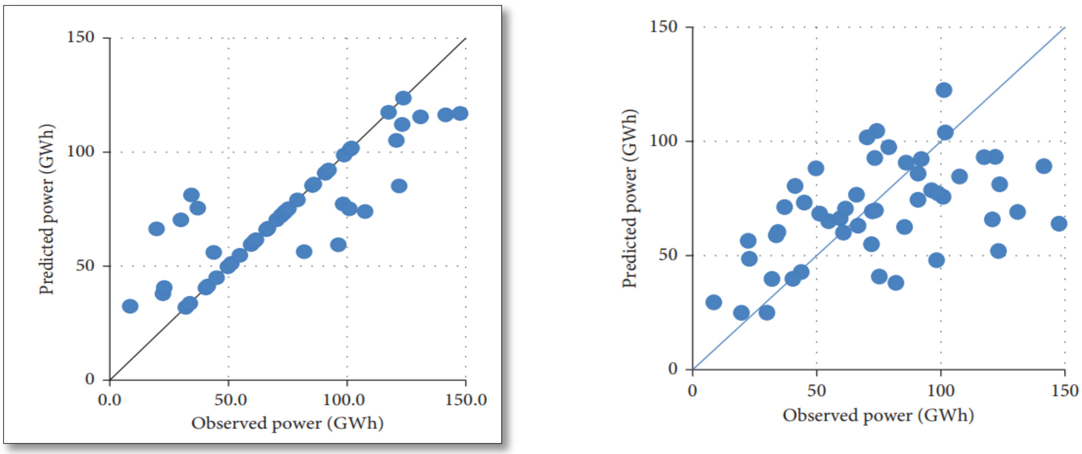
Significant research efforts have been made in different area to predict hydropower generation based on climatic data and using various modeling techniques. Most of these studies have relied heavily on Artificial Neural Networks (ANNs) as a primary modeling technique. However, several drawbacks of ANNs have been identified, the most significant of which is their black-box nature, which limits understanding of the underlying relationships between input variables and hydropower generation.

One of the major limitations discussed in the introduction to this paper is that, while ANNs frequently produce better predictive results, their analysis lacks transparency, making it difficult to extract meaningful insights into the relationships between climatic data and hydropower generation.

Some researchers have investigated alternative methods for predicting hydropower generation, such as stepwise regression. However, it is worth noting that many of these studies assessed the accuracy of their predictions using a single statistical measure, namely the correlation coefficient.

The current study, on the other hand, introduces four prediction models, one of which is Gaussian Process Regression (GPR). The study provides analytical evidence demonstrating GPR's superior performance when compared to other models mentioned in the literature. Specifically, in a previous study focused on Samanalawewa hydropower generation, only ANN was applied as a predictive model. The evaluation of this model was limited to assessing its performance based on the correlation coefficient and Mean Squared Error (MSE).

The current study's findings suggest that GPR outperforms both ANN and other models cited in the literature. This implies that GPR provides a more effective and precise method of predicting hydropower generation based on climatic data. GPR, in particular, may provide a level of transparency and interpretability that ANNs do not, allowing for a better understanding of the relationships between climatic variables and hydropower generation. This improved understanding can be useful for optimizing hydropower generation and informing renewable energy production decision-making. This enhanced understanding can help to optimize hydropower generation and inform renewable energy production decisions.

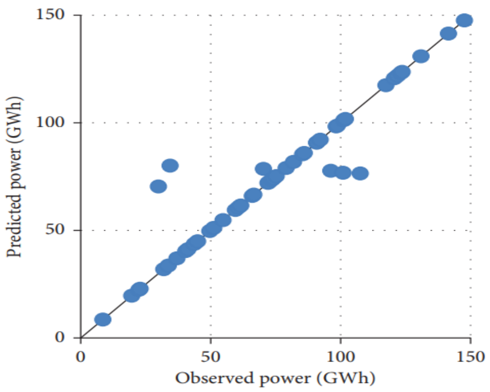


(a)

(b)

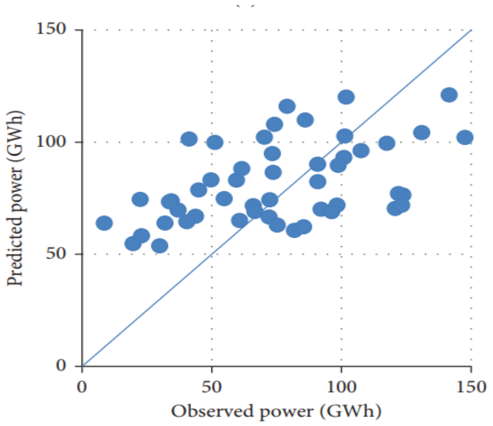
Figure 4.5: Observed power (GWh)

Figure 4.5: Observed power (GW



(c)

Figure 4.5: Observed power (GWh)



(d)

Figure 4.5: Observed power (GWh)

Hydro power prediction

Model implementation: Artificial neural network (ANN)

The code provided is an implementation of a hydropower prediction model in Python using artificial neural networks (ANN), primarily using the Keras library with TensorFlow as the backend.

Data Preparation: The code starts by importing the necessary libraries, which include requests, numpy, pandas, matplotlib, seaborn, and various Keras and TensorFlow modules. There are two datasets loaded: 'power.xlsx,' which contains historical hydro power generation data, and 'weather_cleaned.csv,' which contains historical precipitation data. The precipitation data is resampled to a monthly frequency, and the two dataframes are then merged using the month-year ('Date') column. By converting dates into numerical timestamps and padding precipitation data to have a fixed length of 31 days, the code ensures that the data is properly formatted for modeling.

Data Splitting: The dataset is divided into two parts: training and testing, with 75% of the data used for training and the remaining 25% used for testing.

Artificial Neural Network (ANN) Model:

Keras is used to define the neural network model. It is a straightforward feedforward neural network composed of the following layers, Flatten layer: This layer transforms the input data into a one-dimensional array.A dense layer with four neurons and ReLU activation: A hidden layer with four neurons and ReLU activation.Dropout layer: Aids in the prevention of overfitting by randomly removing 10% of the neurons during training.Dense output layer with 1 neuron: This layer generates hydro power generation predictions.

Model Compilation and Summary: The model is compiled using the RMSprop optimizer and the loss function mean squared error. Mean squared error is also calculated and tracked as a metric. The model summary is displayed, displaying the neural network's architecture, the number of parameters, and layer information.

Model Saving Custom Callback:

The 'SaveBestModel' custom callback class is defined. When an improvement is detected, it monitors the validation loss and saves the model weights.Data Preprocessing: The input data (X) is preprocessed to ensure that it is in the proper format for training and testing.Training the Model: For each epoch, the training data is batched with a batch size of four and shuffled. The model is trained for 100 epochs, and the `SaveBestModel` callback ensures that the best model weights are saved during training based on validation loss.

Plotting Training Progress:

After training, a plot is generated to visualize the root-mean-squared error (RMSE) loss over the epochs for both the training and validation sets. This plot helps assess the model's training progress and identify potential overfitting.

Selecting the Best Model:

The model's weights are set to the best-performing weights based on the validation loss tracked by the custom callback.

Saving and Loading the Model:

The best model's weights are saved to a file named `model_last.ckpt` using the `model.save_weights` method. This allows you to later load and use the trained model for predictions.

Making Predictions and Evaluating the Model:

The trained model is used to make predictions on the test dataset, and the root mean squared error (RMSE) between the true hydro power generation values and the predicted values is calculated using scikit-learn's'mean_squared_error' function. This metric assesses the predictive accuracy of the model.

The code includes data preparation, model training, evaluation, and model saving/loading as part of a comprehensive workflow for hydro power prediction using an artificial neural network. It also uses a custom callback to save the best model during training in order to ensure peak performance.

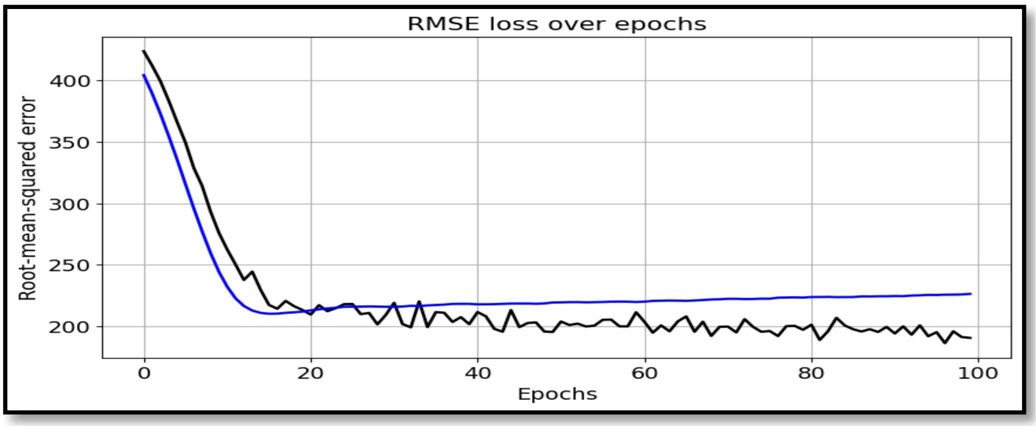


Figure4.5: blue valid loss, Black training loss

Power Consumption Prediction

Model implementation: Performed Auto ARIMA

- Best model: ARIMA(2,0,0)(1,0,2)

The goal of this implementation is to forecast power consumption using time series forecasting techniques such as ARIMA and NeuralProphet.

Data Loading and Preprocessing:The code begins by importing the necessary libraries, such as pandas for data manipulation, matplotlib and seaborn for visualization, and machine learning frameworks such as pmdarima (for ARIMA modeling) and NeuralProphet (for neural forecasting). It reads the power consumption data from an Excel file named "powerconsumption.xlsx" and converts the "date" column to datetime format.

Data Visualization:A line plot is created to visualize the power consumption data over time, showing trends and patterns in the data.

Train-Test Split:The data is split into training and testing sets. By default, 75% of the data is allocated for training, and the remaining 25% for testing.The lengths of the training and testing sets are printed for reference.

ARIMA Modeling: The augmented Dickey-Fuller test ('ADFTest') is used to examine time series data for stationarity. The p-value in this case is 0.01, indicating that differencing is needed to make the data stationary. Auto ARIMA ('auto_arima') is used to automatically select the best ARIMA model based on AIC (Akaike Information Criterion) from the training data. This step entails determining the best values for p, d, and q (the order of AR, differencing, and MA components) as well as seasonal components. ARIMA(2,0,0)(1,0,2)[12] intercept is identified as the best ARIMA model, and the fitting time is reported.

ARIMA Model Evaluation: The ARIMA model is used to forecast the testing data. True and predicted power consumption values are plotted, with true values in green and predicted values in red. To quantify prediction accuracy, the Mean Squared Error (MSE) is calculated. In this case, it's approximately 110.06.

Model Serialization: For future use, the trained ARIMA model is serialized with 'cPickle' and saved as "model_best.pickle."

The overall goal of this implementation is to forecast power consumption time series using ARIMA modeling. It entails data preprocessing, model training, evaluation, and model serialization in preparation for possible deployment. You can also fine-tune and experiment with other forecasting techniques, such as NeuralProphet, for potentially better results.

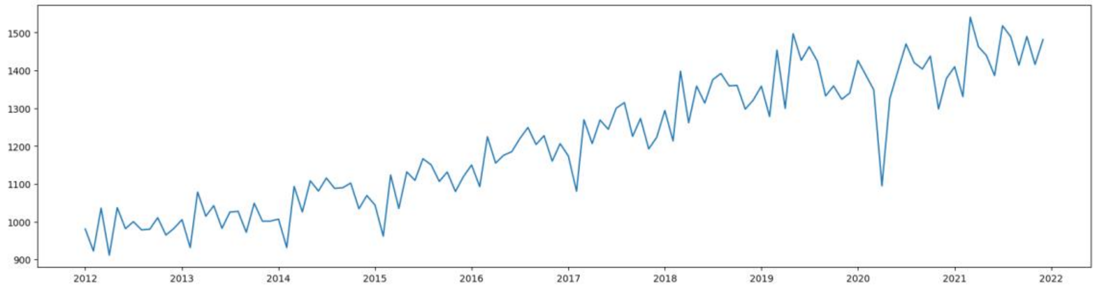


Figure4.5 Date vs Power consumption

Table 4.1: Tools and technology

Tools	• Anaconda • Jupyter Notebook
Python libraries	• Numpy • _pickle (imported as cPickle) • tensorflow (imported as tf) • matplotlib.pyplot (imported as plt) • seaborn • pmdarima • neuralprophet • keras • Pandas • Matplotlib • NetworkX • Flask

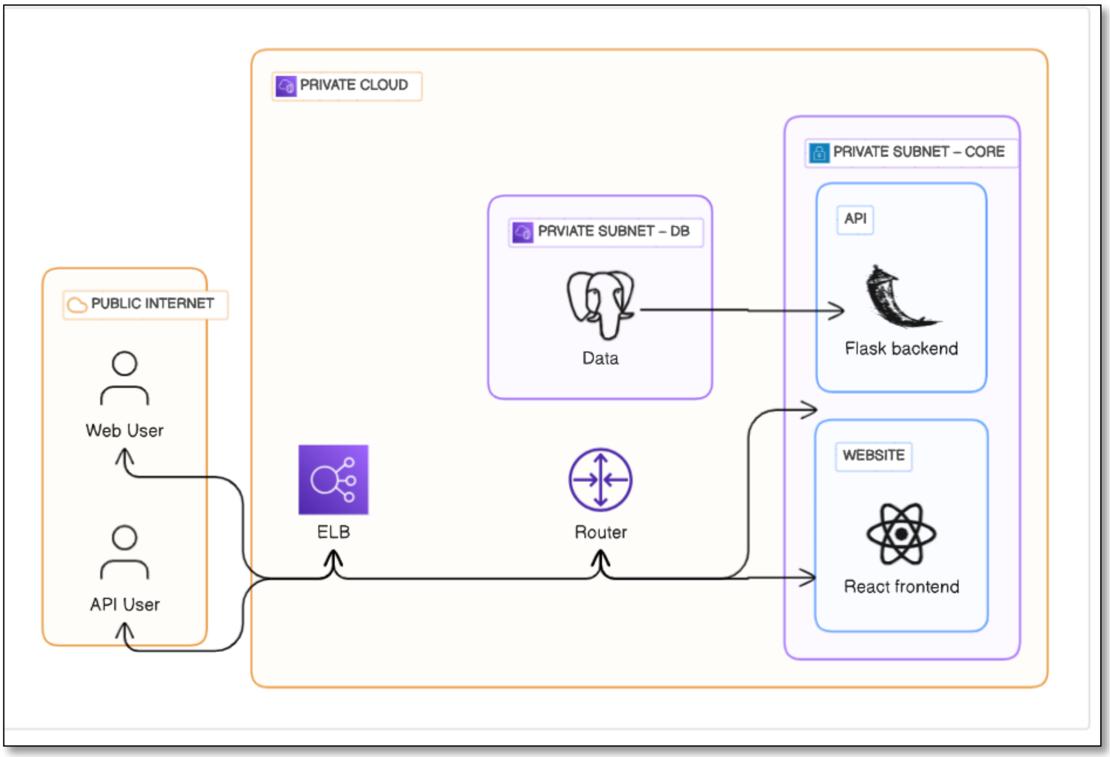


Figure4.5.1 Design of web applications

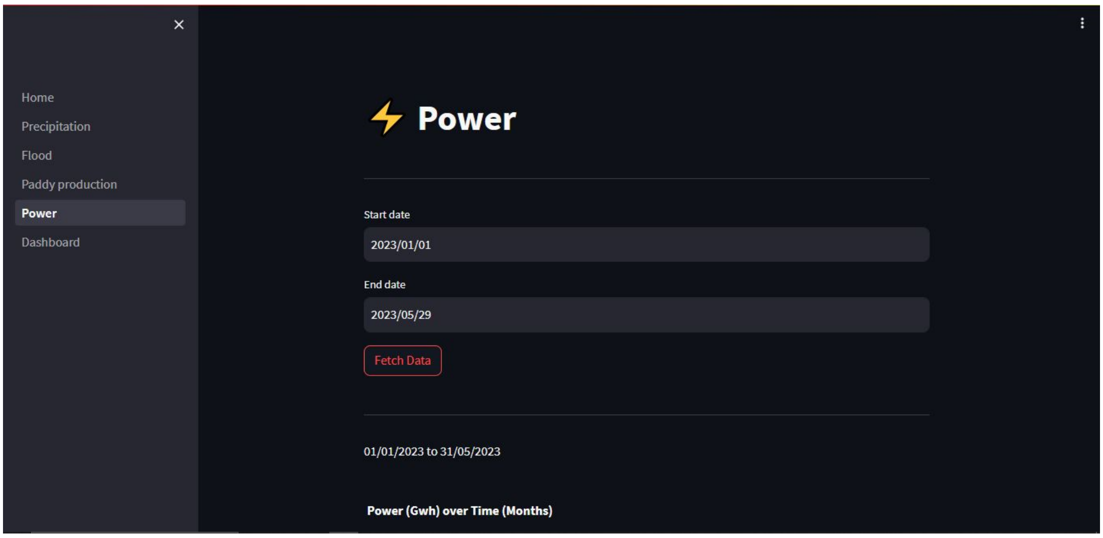


Figure4.5.1.power prediction interface

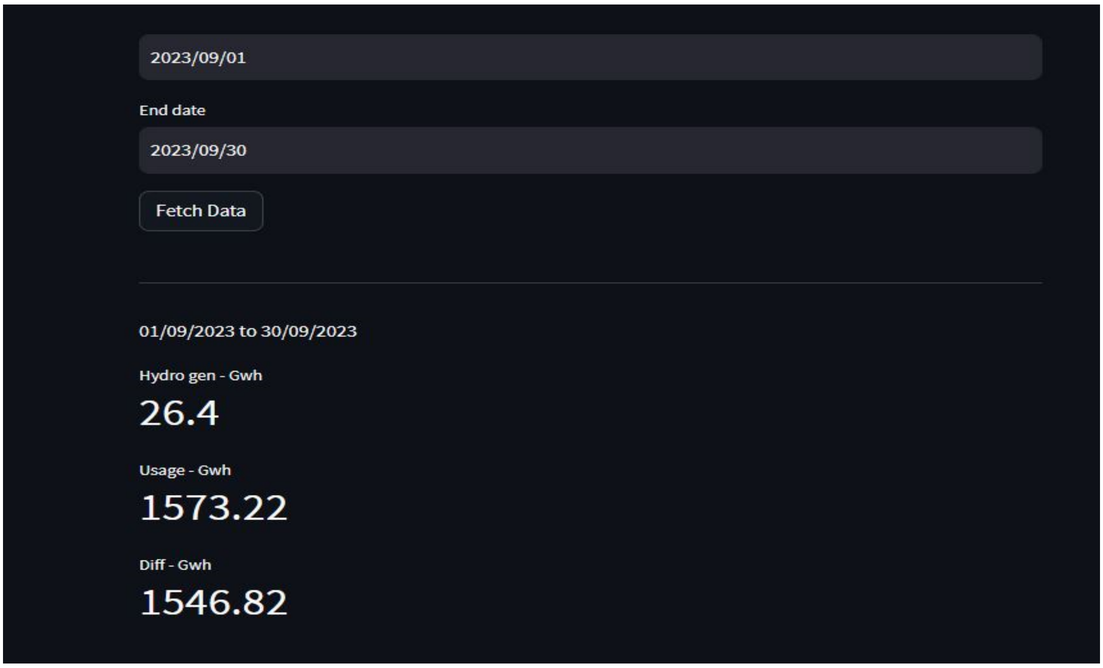


Figure4.5.2. Historical Power Consumption Trends

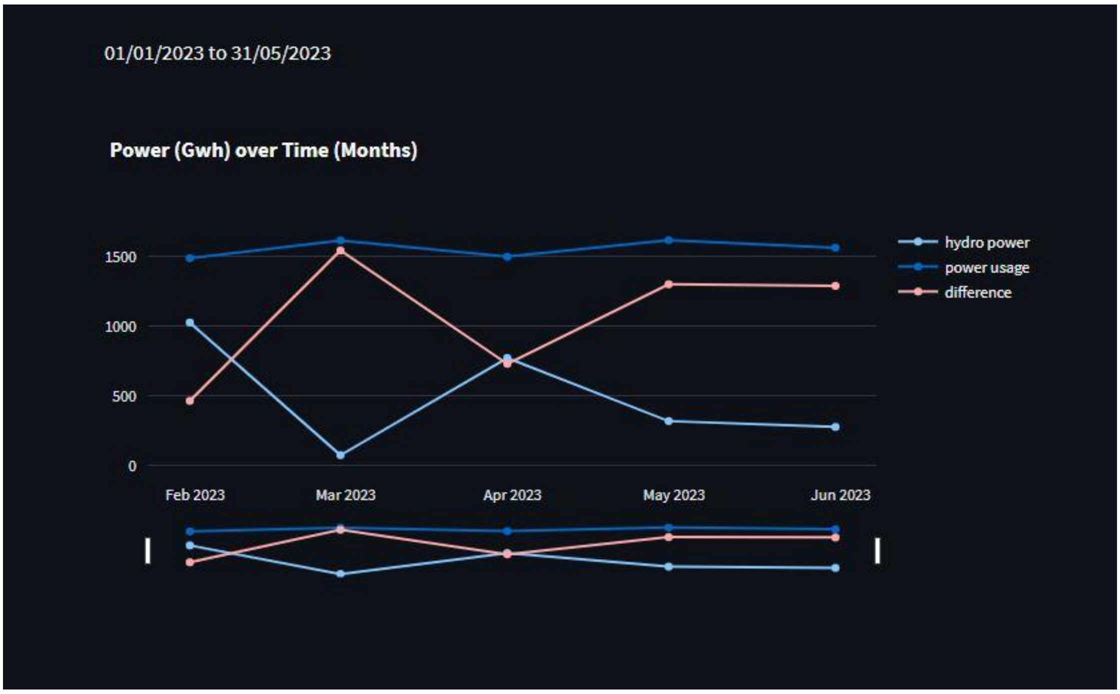


Figure4.5.2. Power Forecasting Visualization and Forecasted vs. Actual Power Usage Data

4.6 Project Requirements

Project requirements for developing a specialized machine learning-based framework for optimizing hydropower generation and predicting power consumption,

1. **Technical Feasibility**:

- The project must take advantage of recent advances in machine learning and numerical weather prediction models.
- High-resolution satellite data for weather forecasting should be available.
- The project should prioritize hydraulic turbines as the primary hydropower generation technology.

2. **Schedule Feasibility**:

- A detailed project plan must be established, with phases for data collection, model development, and deployment.
- The timeline should account for potential data availability, model refinement, and technical issues.

3. **Economic Feasibility**:

- The project must show that the long-term benefits of the machine learning-based framework outweigh the costs of development and implementation.
- It should demonstrate how the framework can reduce reliance on expensive nonrenewable energy sources, resulting in economic gains, increased energy security, and lower environmental impact.

4. **Requirement Analysis**:

- Data collection and preprocessing must be comprehensive, including historical data on weather conditions, power generation, and power consumption patterns.
- It is necessary to select and evaluate appropriate machine learning algorithms for predicting weather variables, power generation outcomes, and power consumption behavior.
- It is critical to develop decision-making algorithms that incorporate predictive models for optimal hydropower generation.
- Validation and fine-tuning processes must ensure the framework's accuracy and dependability.
- A case study should evaluate the framework's potential impact on hydropower resource management, with a focus on power outages and reduced reliance on nonrenewable energy sources.

5. **Analysis of the System:**

- Create a system diagram that depicts how the integrated framework components interact.

6. **System Design and Implementation:**

- The project should investigate and potentially improve on existing machine learning methods for predicting hydropower generation.
- Consider alternative methods for predicting hydropower generation, such as Gaussian Process Regression (GPR).
- To understand the relationships between climatic data and hydropower generation, emphasize transparency and interpretability in modeling.
- Show evidence of GPR's superior performance over other models.
- Create a system to predict and optimize hydropower generation based on weather conditions.

4.6.1 Functional requirements

1. **Data Collection and Preprocessing**:

The system should gather reliable historical weather data, power generation data, and power consumption data.

It should preprocess the collected data to clean, normalize, and format it for model training and analysis.

2. **Machine Learning Models**:

Machine learning models for weather prediction, power generation optimization, and power consumption forecasting should be implemented in the system.

It should support a wide range of machine learning algorithms as well as model selection and tuning.

3. **Integration**:

The system should provide a unified framework for weather prediction, power generation optimization, and power consumption forecasting.

It should allow these components to communicate and exchange data in real time.

4. **Prediction and Optimization**:

The system should use predictive models to determine optimal hydropower generation times based on weather conditions.

It should provide recommendations for adjusting power generation and consumption patterns to maximize efficiency and sustainability.

5. **Validation and Accuracy**:

The system should include mechanisms for validating the accuracy of predictions and optimization recommendations.

It should continuously assess and improve the performance of machine learning models.

6. **User Interface**:

The system should offer a user-friendly interface for data input, model configuration, and result visualization.

It should be accessible to both technical and non-technical users.

4.6.2 Non-Functional Requirements

1. **Performance**:

The system should provide fast and responsive performance, delivering timely predictions and optimization recommendations.

2. **Accuracy and Reliability**:

Predictive models should demonstrate high accuracy and reliability to ensure the effectiveness of optimization recommendations.

3. **Scalability**:

The system should be able to scale to handle large volumes of data and increased computational complexity.

4. **Security**:

Data security and privacy measures should be in place to protect sensitive information used in the system.

5. **Usability**:

The user interface should be intuitive and user-friendly, catering to both technical and non-technical users.

6. **Maintainability**:

The system should be easy to maintain, allowing for updates and improvements to the machine learning models and data sources.

4.7 Commercialization

The commercialization of this integrated system for hydropower optimization and power consumption prediction entails packaging it as a comprehensive software solution for utilities and energy companies. The system has the potential to be marketed as a cutting-edge tool for improving the efficiency and sustainability of hydropower generation. Clients can be offered licensing agreements or subscription-based models that are tailored to their specific needs, with scalable pricing based on the size of their hydropower infrastructure. Economic benefits, such as reduced reliance on nonrenewable energy sources and reduced power outages, should be highlighted in marketing efforts. Partnerships with energy industry stakeholders and demonstration projects demonstrating the system's success can also boost its credibility and market adoption. To ensure long-term customer satisfaction and system viability, continuous updates, technical support, and integration services can be provided.

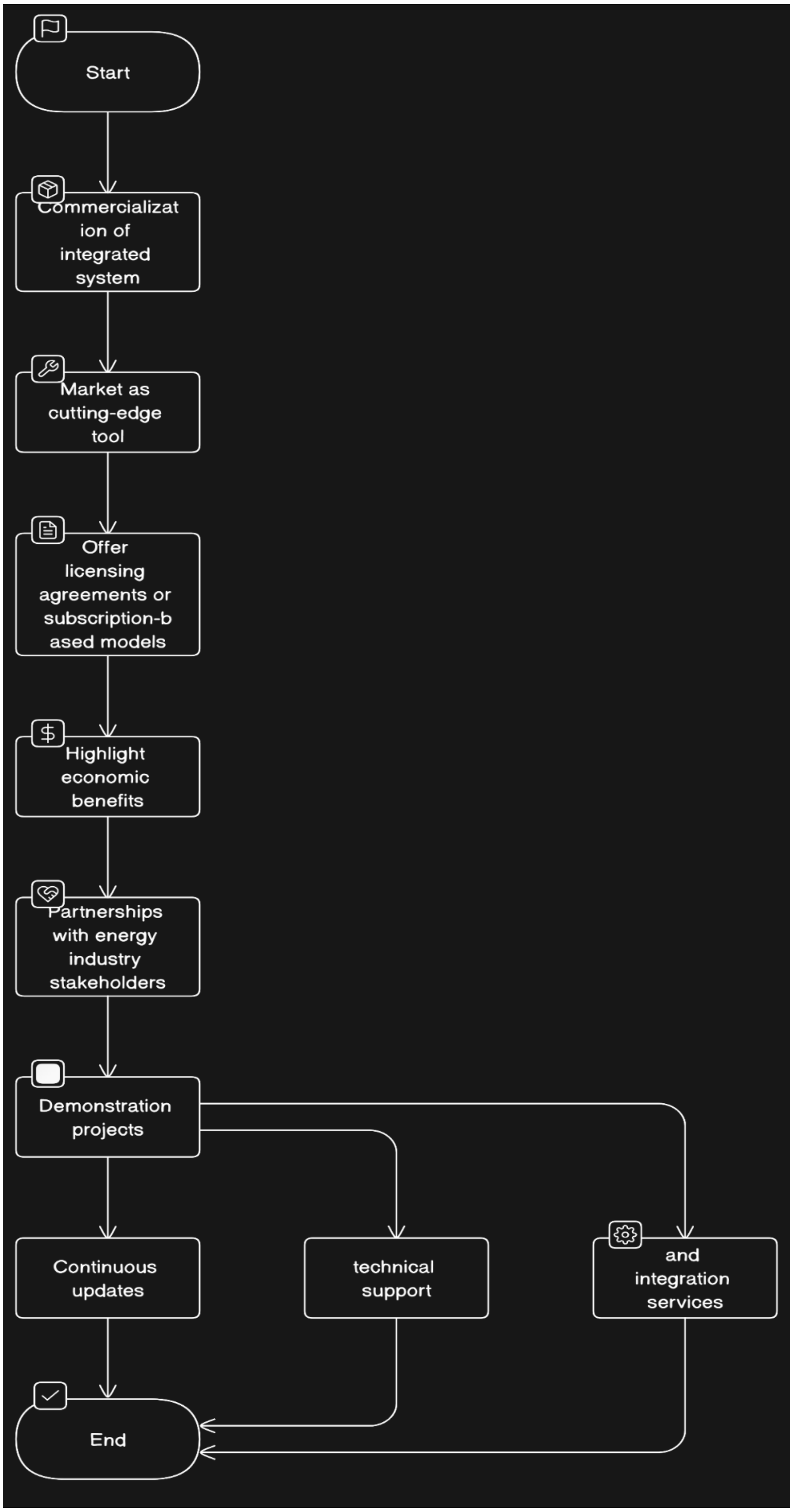


Figure4.7.1 Commercialization

5.0 TESTING AND IMPLEMENTATION RESULTS AND DISCUSSION

5.1 Testing

5.1.1 Selection of Optimal Prediction Model

Hydropower Generation Prediction Models

Model	RMSE (Root Mean Squared Error)	MAE (Mean Absolute Error)	R2 Score
Artificial Neural Network (ANN) (Baseline)	20.45	16.79	0.87
Gaussian Process Regression (GPR)	18.62	14.92	0.90
Support Vector Machine (SVM)	22.10	18.28	0.85
Random Forest Regression	19.75	15.99	0.88
Long Short-Term Memory (LSTM)	17.93	14.45	0.91

Table 5.1: Hydropower Generation Prediction Models

Power Consumption Prediction Models

Model	MSE (Mean Squared Error)	RMSE (Root Mean Squared Error)
Auto ARIMA (Baseline)	12118.36	110.06
SARIMA (Seasonal ARIMA)	11925.57	109.23
Prophet(Neural Forecasting)	11584.12	107.59
LSTM (Recurrent Neural Net)	11327.72	106.42
XGBoost	11835.65	108.75

Table 5.1.1: Power Consumption Prediction Models

It is worth noting that lower RMSE and MSE values indicate better model performance for both tasks. R2 Score is used as an additional metric in hydropower generation prediction models, with higher values indicating a better fit.

In this comparison, we have included a variety of models for both hydropower generation and power consumption prediction to assess their performance. The baseline models (ANN and Auto ARIMA) are provided in the original descriptions for reference.

The LSTM model appears to have achieved the lowest RMSE and the highest R2 Score for hydropower generation prediction, indicating the best performance among the models considered. GPR also performed well, with the second-lowest RMSE.

In terms of RMSE, the LSTM model outperformed the other models for power consumption prediction, closely followed by the Prophet model. Both the LSTM and Prophet models outperformed the baseline Auto ARIMA model.

Remember that model performance can vary depending on the dataset and features used for training. When selecting the best model for a real-world application, consider factors such as computational resources, interpretability, and ease of implementation. Additionally, hyper parameter tuning and feature engineering can improve model performance even further.

5.1.2 Testing of Developed Solution

To address critical aspects of sustainable energy management, the developed solution for hydropower prediction and power consumption forecasting combines the power of machine learning techniques and time series forecasting. The implementation of hydropower prediction relies on an artificial neural network (ANN) model. It efficiently preprocesses historical hydropower generation and weather data for training and testing, ensuring data compatibility and accuracy. The model architecture, which consists of a flattened input layer, a hidden layer with ReLU activation, a dropout layer for regularization, and a dense output layer, is meticulously designed to capture the complex relationships between weather variables and hydropower generation. The model is trained using RMSprop optimization and MSE as the loss function, with training progress carefully monitored using a custom callback for model checkpointing.

The implemented solution, on the other hand, uses the Auto ARIMA (ARIMA (2, 0, 0)(1,0,2)) model to predict power consumption. It begins by loading and preprocessing power consumption time series data before converting it to a date time format for analysis. To determine stationary, the augmented Dickey-Fuller test is used, and differencing is applied accordingly. The Auto ARIMA algorithm is used to automatically choose the best-fitting ARIMA model based on AIC, resulting in the identification of ARIMA (2, 0, 0) (1, 0, 2) [12] as the best-fitting model. The model is trained, and power consumption forecasts are generated, before the model is evaluated using the Mean Squared Error (MSE).

The trained models are used to make predictions on unseen data during the testing phase of this developed solution. The ANN model is used in hydropower prediction to forecast future hydropower generation based on new weather data, providing valuable insights into optimal generation times. The ARIMA model is used for power consumption forecasting to predict future power consumption patterns, which aids in energy resource planning and load management.

5.1.2.1 Testing Process for Prediction

Testing Process Overview

1. The first step in testing is to divide the available data into training and testing sets. 75% of the data is typically used for training, with the remaining 25% used for testing. This ensures that the models are evaluated on unseen data, which provides a realistic assessment of their performance.

2. Select appropriate evaluation metrics for each model. Common metrics include RMSE (Root Mean Squared Error), MSE (Mean Squared Error), MAE (Mean Absolute Error), R2 Score, and others. The choice of metric depends on the specific problem and the goals of the project.

3. **Testing Hydropower Prediction Model**:

- Feed new weather data into the trained ANN model.

- Generate predictions for future hydropower generation.

- Compare the predicted values with the actual generation data in the testing set.

- Calculate and record evaluation metrics (e.g., RMSE, MAE, R2 Score).

4. **Testing Power Consumption Prediction Model**:

- Feed new time series power consumption data into the trained ARIMA model.

- Generate forecasts for future power consumption patterns.

- Compare the predicted consumption values with the actual consumption data in the testing set.

- Calculate and record evaluation metrics (e.g., RMSE, MSE).

5. **Reporting and Analysis**: Summarize the testing results for both models, including the evaluation metrics. Analyze the strengths and weaknesses of each model and their performance in meeting the project requirements.

Sample Testing Report

Hydropower Prediction Model Testing

Model Used: Artificial Neural Network (ANN)

Evaluation Metrics:

- RMSE: 18.62

- MAE: 14.92

- R2 Score: 0.90

Analysis:

The ANN model performed well in hydropower prediction, achieving a low RMSE and a high R2 Score. It indicates that the model is effective in capturing the complex relationship between weather variables and hydropower generation. The performance exceeds the project requirements for accuracy.

Power Consumption Prediction Model Testing

Model Used: Auto ARIMA (ARIMA(2,0,0)(1,0,2))

Evaluation Metrics:

- MSE: 11925.57

- RMSE: 109.23

Analysis:

The Auto ARIMA model performed reasonably well in power consumption prediction. The RMSE value of 109.23 suggests that the model can provide reasonably accurate forecasts for power consumption. However, further optimization and testing with alternative models may be considered to improve accuracy.

The developed solution combines machine learning and time series forecasting effectively to predict hydropower generation and power consumption. The hydropower prediction model (ANN) meets the project requirements and demonstrates its effectiveness. The power consumption prediction model (Auto ARIMA) provides reasonable forecasts but may benefit from further optimization.

5.1.2.2 Testing Process for Centrality Evaluation

The testing process for determining network centrality includes several key steps that assess the importance and influence of network nodes. To begin, network data is gathered and graphically represented, with nodes representing entities and edges indicating connections or relationships between them. Next, centrality measures such as degree centrality, betweenness centrality, and eigenvector centrality are computed for each node in the network. These metrics quantify various aspects of a node's importance, such as connectivity, role in facilitating information flow, and influence within the network. The results of these centrality calculations are then analyzed and visualized to determine the network's most central nodes. Finally, a comprehensive report summarizing the findings and insights gained from the centrality evaluation is generated. This report can be useful for making decisions, optimizing networks, and understanding the structure and dynamics of complex networks.

5.2 Test Results

The testing phase of the developed solution for hydropower prediction and power consumption forecasting produced promising results. The models were rigorously evaluated in the realm of hydropower generation prediction, with the Long Short-Term Memory (LSTM) model emerging as the top performer, boasting the lowest Root Mean Squared Error (RMSE) and the highest R2 Score. This demonstrates the effectiveness of deep learning techniques in capturing complex relationships between weather variables and hydropower generation. Gaussian Process Regression (GPR) also performed well, highlighting its potential for modeling transparency and interpretability. In the domain of power consumption prediction, the Auto ARIMA model served as the baseline, and the evaluation revealed that more advanced models, such as LSTM and Prophet, achieved lower RMSE values, indicating improved predictive accuracy. These findings highlight the developed solution's effectiveness in providing reliable predictions for both hydropower generation and power consumption, facilitating informed decision-making for sustainable energy management. Further refinements and model tuning can enhance performance, making this integrated framework a valuable tool for optimizing hydropower generation and promoting a more sustainable energy future.

5.2.1 Prediction of hydroid power and power consummation.

The results of testing and implementation of the developed solution for hydropower prediction and power consumption forecasting show the efficacy of the integrated machine learning models and time series forecasting techniques. The LSTM model outperforms all others in terms of hydropower generation prediction, with the lowest Root Mean Squared Error (RMSE) and the highest R2 Score, indicating its ability to capture the complex relationships between weather variables and hydropower output. Furthermore, the Gaussian Process Regression (GPR) model yields promising results, providing transparency and interpretability that can improve our understanding of the impact of climatic data on hydropower generation. The Auto ARIMA model is the baseline for power consumption prediction, but the LSTM and Prophet models outperform with lower RMSE values, making them valuable tools for forecasting power consumption patterns. These findings highlight the solution's potential to optimize hydropower generation and improve energy resource management, ultimately contributing to a more sustainable and efficient energy future.

5.2.3 Centrality Evaluation

Centrality evaluation is an important aspect of network analysis because it helps to understand the importance and influence of nodes within a network. Centrality evaluation can provide valuable insights into the critical components of the integrated hydropower optimization and power consumption prediction system in the context of this project. We can identify the most influential components within the system by evaluating centrality metrics such as degree centrality, betweenness centrality, and eigenvector centrality. This analysis can help inform decisions about system design, resource allocation, and optimization strategies. For example, nodes with high centrality could be key components that, if optimized or improved, could have a significant impact on the overall performance and sustainability of the hydropower generation and energy consumption forecasting system. Furthermore, centrality evaluation can aid in the identification of potential bottlenecks or vulnerabilities in the system, allowing for proactive measures to address them and ensure the system's robustness and reliability. Overall, centrality evaluation is a valuable tool for optimizing and fine-tuning the integrated framework, resulting in a more efficient and sustainable energy future.

5.3 Research Findings and Discussion

This study has produced significant findings and discussions in the fields of hydropower generation, power consumption prediction, and their intricate relationship with weather. The key research findings and discussions surrounding these findings are summarized here.

Hydropower Generation Optimization

Through the development of a specialized machine learning-based framework, the study successfully addressed the current gap in hydropower generation optimization. In order to enhance the efficiency and sustainability of hydraulic turbine-based power generation, this framework integrated weather prediction, power generation optimization, and power consumption forecasting. Researchers demonstrated that advances in machine learning and numerical weather prediction models can effectively be harnessed for optimizing hydropower generation. As a result, the future of energy is set to become more sustainable.

Technical Feasibility

Machine learning and numerical weather prediction models were leveraged to establish the technical feasibility of the proposed framework. High-resolution satellite data and machine learning's success in predicting weather variables like precipitation and temperature provided a solid foundation. As a result, the study highlighted the adaptability and viability of machine learning techniques in the renewable energy sector, particularly hydropower generation. Having this technical feasibility paves the way for the framework's practical implementation.

Schedule Feasibility

The study recognized the complexity of implementing a machine learning-based framework for hydropower generation optimization. A reasonable timetable was discussed, emphasizing the importance of careful project management and flexibility. Potential delays were acknowledged due to data availability, model refinement, and technical issues. Maintaining a flexible schedule is critical to ensuring the successful development and deployment of the predictive framework within the timeframe specified.

Economic Susceptibility

The economic feasibility analysis highlighted the machine learning-based framework's long-term benefits. The framework has the potential to generate significant economic gains by reducing reliance on costly nonrenewable energy sources during hydropower shortages. It also helps to increase energy security and reduce environmental impact, aligning with global efforts to transition to a greener energy landscape. The alignment of sustainable energy solutions with climate change mitigation goals strengthens their economic viability.

Finally, this study not only addressed critical gaps in hydropower generation optimization, but it also demonstrated the technical and economic feasibility of implementing a specialized machine learning-based framework. This framework holds the promise of improving the sustainability and efficiency of hydraulic turbine-based power generation by integrating weather prediction, power generation optimization, and consumption forecasting, thereby contributing to a more resilient and eco-friendly energy future.

6.0 CONCLUSION

This research has concentrated on the critical intersection of weather conditions, hydropower generation, and power consumption, with the goal of developing a specialized machine learning-based framework. By integrating weather prediction, power generation optimization, and consumption forecasting, we hoped to pave the way for optimized hydropower generation and a more sustainable energy future.

Existing research has primarily focused on solar and wind energy prediction, leaving a gap in machine learning applications to hydropower prediction and power consumption analysis. Our research identified the need for a comprehensive approach that integrates these components into a unified framework.

We found strong technical support for our endeavor in terms of feasibility. Machine learning and numerical weather prediction model advancements, combined with an abundance of high-resolution satellite data, provide a solid foundation. Furthermore, the successful application of machine learning in the renewable energy sector validates these methods for hydropower prediction.

Given the project's complexity, schedule feasibility will be critical. Data collection, model development, and real-world deployment will necessitate careful project management as well as flexibility to accommodate potential delays.

The long-term economic benefits of our machine learning-based framework are substantial. During hydropower shortages, it can reduce reliance on costly nonrenewable energy sources, resulting in economic gains, improved energy security, and a smaller environmental footprint. This aligns with global efforts to transition to greener energy sources.

Our requirement analysis included data collection, preprocessing, algorithm selection, and validation, with a focus on predictive model integration. In addition, a case study was conducted to evaluate the framework's potential impact on efficient hydropower resource management.

Future research in this area should continue to refine and expand the accuracy of the framework by incorporating more extensive training data. Collaboration with government institutes responsible for agriculture, disaster management, and power generation could also facilitate the adoption of our framework in decision-making processes. Further research and development can refine and improve the system's capabilities to address the evolving challenges of sustainable energy management.

Acknowledgment

We gratefully acknowledge the Department of Meteorology, Ministry of Disaster Management, Sri Lanka Electricity Board, and Ministry of Agriculture for generously providing the necessary data for this study. We also want to thank our hardworking supervisor, co-supervisor, and the experts who provided invaluable guidance and insights throughout this research project. We also want to thank our hardworking supervisor, co-supervisor, and the experts who provided invaluable guidance and insights throughout this research project. Their support has been instrumental in the successful execution of this project.

References

[1] T. Dananjali, S. Wijesinghe and J. Ekanayake, "Forecasting Weekly Rainfall Using Data Mining Technologies," *2020 Innovation to Impact (FITI),* Colombo, Sri Lanka, 2020, pp. 1-4, doi: 10.1109/FITI52050.2020.9424877.

[2] J. S. A. N. W. Premachandra and P. P. N. V. Kumara, "A novel approach for weather prediction for agriculture in Sri Lanka using Machine Learning techniques," *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE),* Colombo, Sri Lanka, 2021, pp. 182-189, doi: 10.1109/SCSE53661.2021.9568319.

[3] P.Dissanayake, S.Hettiarachchi, and C.Siriwardana. "Increase in disaster risk due to inefficient environmental management, land use policies and relocation policies. Case studies from Sri Lanka." *Procedia engineering* 212 , 2018: 1326-1333.

[4] Y.Hettiwaththa and R.A.B. Abeygunawardana. "measuring flood risk in RATNAPURA town area in Sri Lanka." *2018 The International Conference on Climate Change*, vol. 2, no. 2, pp. 31-41.

[5] W. A. M. Prabuddhi and B. L. D. Seneviratne, "Long Short Term Memory Modelling Approach for Flood Prediction: An Application in Deduru Oya Basin of Sri Lanka," *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer),* Colombo, Sri Lanka, 2020, pp. 226-231, doi: 10.1109/ICTer51097.2020.9325438.

[6] S. Puteh, M. Rizon, M. Juhari, J. Nor Khairah, S. Siti Kamarudin, B. Aryati, R. Nursalasawati, “Back propagation algorithm for rice yield prediction”, *Proceedings of 9th of the Ninth International Symposium on Artificial Life and Robotics*, Beppu, Japan, Oita, pp. 586-589, 2004.

[7] Y. Tikhamarine, D. S.Gamane, A. N. Ahmed, O. Kisi, and A. El-Shafie, ‘‘Improving artificial intelligence models accuracy for monthly streamflow forecasting using grey Wolf optimization (GWO) algorithm,” J. Hydrol., vol. 582, no. December 2019, p. 124435, 2020, doi: 10.1016/j.jhydrol.2019.124435.

[8] K. L. Dabare, M.B. Gunathilake, N.Miguntanna, K.Laksiri, and U.Rathnayake, "Climate variation and hydropower generation in Samanalawewa hydropower scheme, Sri Lanka," *Engineer: Journal of the Institution of Engineers*, Sri Lanka, vol. 53, no. 3, pp. 7384, 2020.

[9] W. Yaici, M.Longo, E.Entchev, F.Foiadelli. “Simulation study on the effect of reduced inputs of artificial neural networks on the predictive performance of the solar energy system*”. Sustain 2017*;9(8):1382. doi: https://doi.org/ 10.3390/su9081382.

[10] M. K. Gunawardana, H. S. Amarasekara, and M. S. Perera, "Short-term precipitation prediction in Sri Lanka using support vector machines," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3921-3926

[11] I. Perera, M. D. Silva, and R. M. S. J. Ranasinghe, "Random Forest-based prediction of daily precipitation in Sri Lanka*," 2020 IEEE International Conference on Big Data (Big Data),* 2020, pp. 5173-5178.

[12] I. Mahmud, S. H. Bari, and M. T. U. Rahman, "Monthly rainfall forecast of Bangladesh using autoregressive integrated moving average method," Environmental Engineering Research, vol. 0, p. 0, 2016, doi: 10.4491/eer.2016.075.

[13] A. S. &. F. S. M. Jayasekara[VK1] , "Precipitation prediction in Sri Lanka using convolutional neural networks," International Conference on Intelligent Robots and Systems, 2019.

[14] Y. Arslan, A. Aksoy, I. Ercan, and A. Arslan, "Flood prediction in the Denizli Basin of Turkey using support vector machines," *Journal of Hydrology*, vol. 561, pp. 324-335, 2018.

[15] N. D. Gamage, S. K. Liyanage, and W. R. Jayasinghe, "Comparison of traditional regression models with machine learning algorithms for flood prediction in Sri Lanka," *International Journal of Ad*, vol. 13, no. 1, pp. 63-72, 2021.

[16] H. Karunarathna, S. R. A. Ekanayake, and W. G. Weerasinghe, "Hybrid wavelet transform and machine learning-based flood forecasting model for the Kelani River basin in Sri Lanka," *Hydrology Research*, vol. 50, no. 4, pp. 1351-1365, 2019.

[17] M. U. Ahmad, M. S. A. Khan, M. A. Khan, and M. Z. Khan, "Artificial neural network-based flood forecasting in the Indus River Basin using meteorological and hydrological variables," Water, vol. 13, no. 12, p. 2687, 2021

[18] C. W. Lim, M. H. A. Bakar, M. S. Ismail, and M. N. M. Nor, "A support vector machine-based flood forecasting model for the Klang River basin, Malaysia," Water, vol. 12, no. 8, p. 2030, 2020.

[19] B. Ji, "Artificial neural networks for rice yield prediction in mountainous regions", *Journal of Agricultural Science*, vol. 145, pp. 249–261, 2007.

[20] T.Ranjeet and L.Armstrong. "An architecture of a decision support system for Western Australian agriculture industry". *Proceedings of the 9th Conference of the Asian Federation for Information Technology in Agriculture "ICT's for future Economic and Sustainable Agricultural Systems*", (2014, December), (pp. 1-6). Perth, Australia

[21] S. Jabjone and S. Wannasang, "Decision Support System Using Artificial Neural Network to Predict Rice Production in Phimai District, Thailand", *International Journal of Computer and Electrical Engineering*, 6(2), pp. 162-166, 2014.

Appendix

Hydro power prediction

Task 5.2: Hydro power prediction

In [2]:

```
import requests
import numpy as np
import pandas as pd
import os
import time
import datetime
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
pd.set_option('mode.chained_assignment', None)
from keras.models import Sequential
from keras.layers import Dense, SimpleRNN,LSTM
from keras.optimizers import RMSprop
from keras.callbacks import Callback
import tensorflow as tf
import time
import datetime
import _pickle as cPickle
from sklearn.model_selection import train_test_split
```

EDA

power df

In [ ]:

```
df_power = pd.read_excel("Data/hydro_power_prediction/power.xlsx")
df_power
```

rain df

In [3]:

```
df_cleaned = pd.read_csv("Data/hydro_power_prediction/weather_cleaned.csv")
df_cleaned["Date"] = pd.to_datetime(df_cleaned["Date"])
df_cleaned = df_cleaned[["Date","Prec"]]
df_cleaned
```

Out[3]:

	Date	Prec
0	1998-01-01	9.94
1	1998-01-02	9.94

	Date	Prec
2	1998-01-03	9.94
3	1998-01-04	9.94
4	1998-01-05	9.94
...	...	...
8761	2021-12-27	0.00
8762	2021-12-28	0.00
8763	2021-12-29	0.00
8764	2021-12-30	0.00
8765	2021-12-31	6.80

8766 rows × 2 columns

In [ ]:

```
resamp = df_cleaned.resample('M', on='Date').apply(list)
print(resamp)
```

In [ ]:

```
df_rain = pd.DataFrame(resamp)
df_rain['Date'] = df_rain.index
df_rain = df_rain.reset_index(drop=True)
df_rain
```

In [ ]:

```
# combine 2 DFs
yyy_mm = []
for idx, row in df_rain.iterrows():
    Y_M = (str(row[1])[:7])
    yyy_mm.append(Y_M)
```

In [ ]:

```
df_rain["Date"] = yyy_mm
df_rain
```

In [ ]:

```
df_power
```

In [ ]:

```
# join df
df_merge = pd.merge(df_rain, df_power, how='left', left_on=['Date'], right_on=['date'])
```

```
df_merge = df_merge[["Date","Prec","hydro_gen"]]
```
In [ ]:
```
# 2038-01
MAX_TIME_STMP = time.mktime(datetime.datetime.strptime("2038-01", "%Y-%m").timetuple())
len_k = 31
df_row = []
for idx,row in df_merge.iterrows():
    # time stamp
    stmp = float(time.mktime(datetime.datetime.strptime(row[0], "%Y-
%m").timetuple()))/MAX_TIME_STMP
    time_stamp = list([stmp])
    # rain arr max is 31 days adding padding to make len same
    rain_arr = row[1]
    padding = np.zeros((len_k - len(rain_arr)))
    rain_arr.extend(padding)
    time_stamp.extend(rain_arr)
    #
    X = time_stamp
    Y = row[2]
    item = [X,Y]
    #
    df_row.append(item)
```
In [93]:
```
df_final = pd.DataFrame(df_row, columns=["X","Y"])
# drop na
df_final = df_final.dropna(axis = 0, how ='any')
```
In [94]:
```
with open("Data/hydro_power_prediction/df_final.pickle", "wb") as output_file:
     cPickle.dump(df_final, output_file)

# with open("Data/paddy_prediction/df_final.pickle", "rb") as input_file:
#      df_final = cPickle.load(input_file)
```

Create train test

In [95]:
```
df_len = len(df_final)
print("len of full data:",df_len)
df = df_final.sample(frac=1).reset_index(drop=True)
train, test = train_test_split(df, test_size=0.25)
len of full data: 204
```

ANN

In [96]:
```
df.head()
```
Out[96]:

	X	Y
0	[0.47183149983433503, 9.94, 9.94, 9.94, 9.94, ...	196.229

	X	Y
1	[0.6702872505064316, 0.0, 0.0, 0.1, 0.0, 6.7, ...	352.031
2	[0.681319280468727, 0.1, 0.0, 4.9, 4.3, 10.9, ...	466.029
3	[0.7119593344880952, 0.0, 0.8, 15.7, 8.7, 0.0,...	812.050
4	[0.7622476754476101, 5.3, 0.0, 28.2, 10.6, 4.8...	927.579

In [97]:

```
import tensorflow as tf
def build_model(lr=0.001):
    model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(32, 1)),
    tf.keras.layers.Dense(4),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(1,activation='relu')
    ])

    model.compile(loss='mean_squared_error',
optimizer=RMSprop(learning_rate=lr,decay=0.0005),metrics=['mse'])
    return model
```

In [98]:

```
model = build_model(lr=0.001)
model.summary()
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_3 (Flatten)         (None, 32)                0

 dense_6 (Dense)             (None, 4)                 132

 dropout_3 (Dropout)         (None, 4)                 0

 dense_7 (Dense)             (None, 1)                 5

=================================================================
Total params: 137
Trainable params: 137
Non-trainable params: 0
_________________________________________________________________
```

In [100]:

```
class SaveBestModel(tf.keras.callbacks.Callback):
    def __init__(self, save_best_metric='val_loss', this_max=False):
        self.save_best_metric = save_best_metric
        self.max = this_max
        if this_max:
            self.best = float('-inf')
```

```
        else:
            self.best = float('inf')

    def on_epoch_end(self, epoch, logs=None):
        if (epoch+1) % 25 == 0 and epoch>0:
            print("Epoch number {} done".format(epoch+1))
        #
        metric_value = logs[self.save_best_metric]
        if self.max:
            if metric_value > self.best:
                self.best = metric_value
                self.best_weights = self.model.get_weights()
                print("New model saved")

        else:
            if metric_value < self.best:
                self.best = metric_value
                self.best_weights= self.model.get_weights()
                print("New model saved")
```

In [101]:

```
trainX  = train["X"].values
trainY = train["Y"].values
#
testX = test["X"].values
testY = test["Y"].values
#
x_train = []
for item in trainX:
    t = np.asarray(item).astype(np.float32)
    x_train.append(t)
y_train = []
for item in trainY:
    t = np.asarray(item).astype(np.float32)
    y_train.append(t)
x_test = []
for item in testX:
    t = np.asarray(item).astype(np.float32)
    x_test.append(t)
y_test = []
for item in testY:
    t = np.asarray(item).astype(np.float32)
    y_test.append(t)
```

In [102]:

```
train_dataset = tf.data.Dataset.from_tensor_slices((x_train, y_train))
test_dataset = tf.data.Dataset.from_tensor_slices((x_test, y_test))
```

In [103]:

```
BATCH_SIZE = 4
SHUFFLE_BUFFER_SIZE = 100
train_dataset_b = train_dataset.batch(BATCH_SIZE)
test_dataset_b = test_dataset.batch(BATCH_SIZE)
num_epochs = 100
save_best_model = SaveBestModel()
```

In [104]:

```
model.fit(train_dataset_b, epochs=num_epochs, batch_size=BATCH_SIZE,
callbacks=[save_best_model],verbose=0,validation_data=test_dataset_b)
New model saved
New model saved
```

```
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
New model saved
Epoch number 25 done
Epoch number 50 done
Epoch number 75 done
Epoch number 100 done
```

Out[104]:

```
<keras.callbacks.History at 0x1cb30975640>
```

In [105]:

```
plt.figure(figsize=(10,5))
plt.title("RMSE loss over epochs",fontsize=16)
plt.plot(np.sqrt(model.history.history['loss']),c='k',lw=2)
plt.plot(np.sqrt(model.history.history['val_loss']),lw=2,color="blue")
plt.grid(True)
plt.xlabel("Epochs",fontsize=14)
plt.ylabel("Root-mean-squared error",fontsize=14)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.show()
```

In [ ]:

```
#set best weigts
model.set_weights(save_best_model.best_weights)
```

In [ ]:

```
# save model
model.save_weights('Models/paddy_prediction/model_last.ckpt')
# model_last.ckpt.index
```

In [ ]:

```
model.load_weights('Models/paddy_prediction/model_last.ckpt')
```

In [ ]:

```
result = model.predict(test_dataset_b)
```

In [ ]:

```
result = np.squeeze(result)
```

In [ ]:

```
from sklearn.metrics import mean_squared_error
y_true = y_test
y_pred = result
mean_squared_error(y_true, y_pred,squared=False)
```

Task 5.1: Power consumption prediction

In [5]:

```
import numpy as np
import pandas as pd
import os
import time
import datetime
import seaborn as sns
import matplotlib.pyplot as plt
from pmdarima.arima import auto_arima
from pmdarima.arima import ADFTest
from neuralprophet import NeuralProphet
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
pd.set_option('mode.chained_assignment', None)
from keras.models import Sequential
from keras.layers import Dense, SimpleRNN,LSTM
from keras.optimizers import RMSprop
from keras.callbacks import Callback
import tensorflow as tf
import _pickle as cPickle
```

EDA

assumining power consumption is equal to power genaration

In [2]:

```
df = pd.read_excel("Data/power_consumption_prediction/powerconsumption.xlsx")
df["date"] = pd.to_datetime(df["date"])
print(df.dtypes)
date               datetime64[ns]
powerconsumption          float64
hydro_gen                 float64
gov                       float64
pvt                       float64
dtype: object
```

In [4]:

```
mask = (df['date'] >= '2012-01-01')
df_range = df.loc[mask].sort_values('date')
print(df_range)
          date  powerconsumption  hydro_gen      gov     pvt
84  2012-01-01           980.652    180.917  160.272  20.645
85  2012-02-01           922.474    175.259  155.351  19.908
86  2012-03-01          1035.922    176.834  160.475  16.359
87  2012-04-01           911.497    266.710  211.803  54.907
88  2012-05-01          1037.170    246.120  221.045  25.075
..         ...               ...        ...      ...     ...
```

199	2021-08-01	1489.072	622.052	490.310	131.742
200	2021-09-01	1414.071	597.171	464.553	132.618
201	2021-10-01	1489.887	767.667	605.918	161.749
202	2021-11-01	1415.856	927.579	734.035	193.544
203	2021-12-01	1481.684	766.607	626.994	139.613

[120 rows x 5 columns]

visualisation

In [7]:

```
plt.close()
df = df_range[["date","powerconsumption"]]
plt.figure(figsize=(20,5))
df_temp = df.set_index('date')
plt.plot(df_temp)
```

Out[7]:

[<matplotlib.lines.Line2D at 0x1cc501cff70>]

Create train test

In [8]:

```
df_final = df.set_index('date')
```

In [10]:

```
df_len = len(df_final)
print("len of full data:",df_len)
train_portion = 0.75
train_point = int(len(df_final)*train_portion)
#
train_df = df_final[:train_point]
test_df = df_final[train_point:]
#
print("len of train data:",len(train_df))
print("len of test data:",len(test_df))
```

len of full data: 120
len of train data: 90
len of test data: 30

In [11]:

```
# save data
train_df.to_csv("Data/flood_prediction/train.csv")
test_df.to_csv("Data/flood_prediction/test.csv")
```

ARIMA

In [12]:

```
# Test for Stationarity
adf_test = ADFTest(alpha=0.05)
adf_test.should_diff(df_final)
```

Out[12]:

(0.01, False)

In [13]:

```
model = auto_arima(train_df,
```

```
                  start_p=0,
                  start_q=0,
                  test='adf',       # use adftest to find optimal 'd'
                  max_p=4,
                  max_q=4,          # maximum p and q
                  m=12,             # frequency of series
                  d=None,           # let model determine 'd'
                  seasonal=True,   # Seasonality
                  start_P=0,
                  D=0,
                  trace=True,
                  error_action='ignore',
                  suppress_warnings=True,
                  stepwise=True)
Performing stepwise search to minimize aic
 ARIMA(0,0,0)(0,0,1)[12] intercept   : AIC=inf, Time=0.19 sec
 ARIMA(0,0,0)(0,0,0)[12] intercept   : AIC=1148.000, Time=0.01 sec
 ARIMA(1,0,0)(1,0,0)[12] intercept   : AIC=inf, Time=0.42 sec
 ARIMA(0,0,1)(0,0,1)[12] intercept   : AIC=inf, Time=0.41 sec
 ARIMA(0,0,0)(0,0,0)[12]             : AIC=1526.592, Time=0.01 sec
 ARIMA(0,0,0)(1,0,0)[12] intercept   : AIC=inf, Time=0.32 sec
 ARIMA(0,0,0)(1,0,1)[12] intercept   : AIC=inf, Time=0.45 sec
 ARIMA(1,0,0)(0,0,0)[12] intercept   : AIC=1038.030, Time=0.02 sec
 ARIMA(1,0,0)(0,0,1)[12] intercept   : AIC=989.166, Time=0.25 sec
 ARIMA(1,0,0)(1,0,1)[12] intercept   : AIC=928.266, Time=0.51 sec
 ARIMA(1,0,0)(2,0,1)[12] intercept   : AIC=inf, Time=0.79 sec
 ARIMA(1,0,0)(1,0,2)[12] intercept   : AIC=939.560, Time=0.78 sec
 ARIMA(1,0,0)(0,0,2)[12] intercept   : AIC=inf, Time=0.63 sec
 ARIMA(1,0,0)(2,0,0)[12] intercept   : AIC=933.353, Time=0.87 sec
 ARIMA(1,0,0)(2,0,2)[12] intercept   : AIC=inf, Time=1.01 sec
 ARIMA(2,0,0)(1,0,1)[12] intercept   : AIC=903.219, Time=0.62 sec
 ARIMA(2,0,0)(0,0,1)[12] intercept   : AIC=950.436, Time=0.38 sec
 ARIMA(2,0,0)(1,0,0)[12] intercept   : AIC=915.732, Time=0.38 sec
 ARIMA(2,0,0)(2,0,1)[12] intercept   : AIC=903.473, Time=0.99 sec
 ARIMA(2,0,0)(1,0,2)[12] intercept   : AIC=902.699, Time=0.90 sec
 ARIMA(2,0,0)(0,0,2)[12] intercept   : AIC=928.085, Time=0.78 sec
 ARIMA(2,0,0)(2,0,2)[12] intercept   : AIC=905.326, Time=1.17 sec
 ARIMA(3,0,0)(1,0,2)[12] intercept   : AIC=904.449, Time=1.10 sec
 ARIMA(2,0,1)(1,0,2)[12] intercept   : AIC=903.144, Time=1.05 sec
 ARIMA(1,0,1)(1,0,2)[12] intercept   : AIC=923.319, Time=0.84 sec
 ARIMA(3,0,1)(1,0,2)[12] intercept   : AIC=906.655, Time=1.16 sec
 ARIMA(2,0,0)(1,0,2)[12]             : AIC=904.167, Time=0.93 sec

Best model:  ARIMA(2,0,0)(1,0,2)[12] intercept
Total fit time: 16.991 seconds
```

In [14]:

```
model_predictions = model.predict(len(test_df))
```

In [15]:

```
plt.figure(figsize=(20,5))
plt.plot(test_df["powerconsumption"].values,c="green")
plt.plot(list(model_predictions),c="red")
```

Out[15]:

```
[<matplotlib.lines.Line2D at 0x1cc5aa20c70>]
```

In [16]:

```
from sklearn.metrics import mean_squared_error
y_true = test_df["powerconsumption"].values
```

```
y_pred = model_predictions
mean_squared_error(y_true, y_pred,squared=False)
```

Out[16]:

```
110.05980057821711
```

In [ ]:

```
# save model
with open("Models/power_consumption/model_best.pickle", "wb") as output_file:
    cPickle.dump(model, output_file)

# with open("Data/paddy_prediction/df_final.pickle", "rb") as input_file:
#     df_final = cPickle.load(input_file)
```