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

TMP-23-364

Final Project Thesis

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 acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Student ID	Name	Signature
IT20155148	U.P Hasin Pramodya	*hasin*
IT20161392	K.D. Hasaranga	hasaranga
IT20115784	Y.D. Abeygunawardana	*Yasith*
IT20148836	C.P. Weerasinghe	*chanduka*

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Date: 01/11/2023

ABSTRACT

Sri Lanka mainly experiences two main monsoon seasons called the Southwest Monsoon and the Northeast Monsoon. Rathnapura district takes a prominent place among the areas that receive rain from southwest monsoon. Being located near a mountain face, Ratnapura district receives as well as the convective rains. Ratnapura district is at the top, among the regions with the highest annual rainfall in Sri Lanka. Thus, due to the inability to predict the heavy rainfall in the Ratnapura district in advance, many cases of natural disasters such as floods and the destruction of agricultural crops have been reported. Due to that, accurate and efficient rainfall forecasting is necessary for the Ratnapura district, so this study focuses on regional-based rainfall prediction for the Rathnapura district and predicting the impact of rainfall on floods, agriculture, and others using advanced machine learning techniques.

Acknowledgment

We 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 CONTENT

Table of Contents

1. INTRODUCTION

Rainfall is one of the most important weather conditions in Sri Lanka. Forecasting rainfall can help to solve several problems related to the tourism industry, natural disaster management, agricultural industry, etc. As the Sri Lankan rural economy is mostly based on agriculture, it is important to forecast rainfall as well as other weather conditions accurately [1]. Sri Lanka experiences two main monsoon seasons called the Southwest Monsoon and the Northeast Monsoon. Rathnapura district takes a prominent place among the areas that receive rain from southwest monsoon. Near a mountain face, Ratnapura district receives as well as the convective rains. Ratnapura district is at the top, among the regions with the highest annual rainfall in Sri Lanka. Thus, due to the inability to predict the heavy rainfall in the Ratnapura district in advance, many cases of natural disasters such as floods and the destruction of agricultural crops have been reported. Because of that this study is focused on regional-based rainfall prediction for the Rathnapura district and predicting the impact of rainfall on floods, agriculture, and others.
In Sri Lanka, which is prone to floods, droughts, landslides, and cyclones, flooding emerges as one of the most frequently occurring and damaging natural disasters [3]. It affects human life, the infrastructure, agriculture, and the social and economic systems of a country.

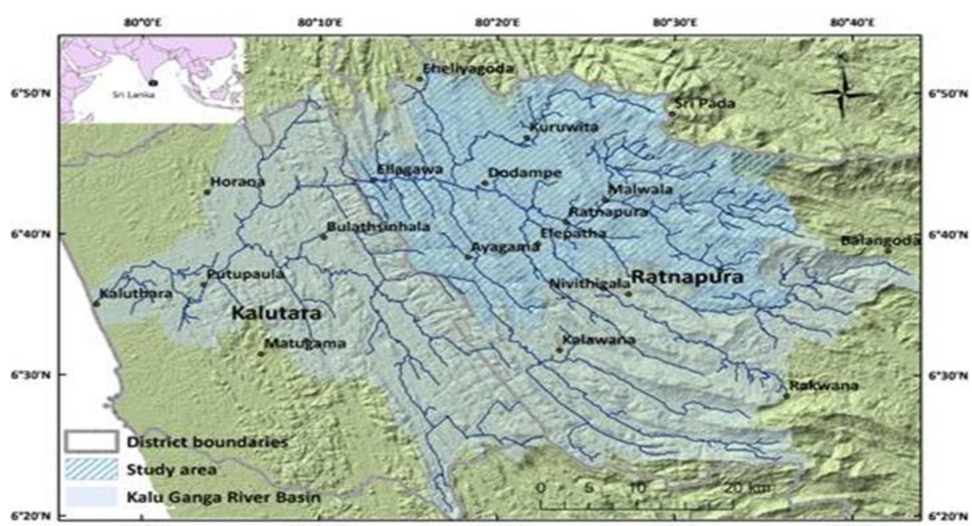


Figure 1.0: Kalu Ganga River basin

According to the Disaster Management Center and the Department of Meteorology of Sri Lanka, heavy rainfall is the main cause of flooding compared to other human activities. Among the affected areas from the flooding, the Rathnapura district stands out as one of the most vulnerable [4]. Because the Kalu Ganga, one of the largest rivers in Sri Lanka, which receives extremely high rainfall and has a high discharge, flows through the Rathnapura District. During the previous 22 years, the Ratnapura district faced the largest number of flood events. So, forecasting flood occurrence is particularly important in most of the applications in disaster management and risk mitigation systems [5]. This study delves into flood forecasting in Rathnapura District, using hydro meteorological data and advanced machine learning techniques, with the intention of improving disaster preparedness and response.

As rice is the staple food in Sri Lanka, rice cultivation has been given high priority in agriculture. However, due to rice use in other food products, rice has become the most popular crop nowadays. Therefore, it is very important to predict the rice yield in different provinces of Sri Lanka. Among the rice-cultivating districts in Sri Lanka, the Ratnapura districts of Sabaragamuwa province occupy a leading position. It is well known that climate change has a significant impact on the cultivation of paddy. Therefore, it is beneficial in many ways to understand the connections between climatic variables and paddy yield. This study uses paddy harvesting data from the Rathnapura District, Sri Lanka, to demonstrate an artificial neural network (ANN) framework that may be utilized to assess the correlations between meteorological factors and the paddy yield. as a most beneficial paddy-producing region in Sri Lanka is a Sabaragamuwa province, thus the research has a lot of promise and interest. In this study, climate variables including rainfall were considered and tried to predict the rice yield in Yala and Maha sessions [6].

In Sri Lanka, power generation encompasses a diverse array of methods to meet the energy demand, yet challenges persist in providing uninterrupted electricity. Variability in sources like hydroelectricity, susceptible to droughts, compels the utilization of alternative energies [7]. Addressing these issues necessitates a comprehensive comprehension of the power system and consumption patterns. By amalgamating the principles of hydropower generation with adaptable energy strategies [8], a continuous and reliable electricity supply becomes feasible. This research embarks on an exploration of this paradigm, emphasizing the cyclic nature of energy consumption and its correlation with climatic conditions. Through predictive modeling of hydropower generation influenced by rainfall, the study seeks to establish a dynamic framework for energy allocation [9]. Consequently, a harmonized system balancing hydropower dependence during the wet season and seamless transition to alternative sources during droughts could revolutionize Sri Lanka's energy landscape, fostering infrastructure development and sustained power availability.

According to the above paragraphs, the Ratnapura district in the Sabaragamuwa province of Sri Lanka has been affected by floods due to rainfall, agricultural crops have been damaged due to adverse weather extremes and has the potential to collect water for hydropower generation. For these reasons, this study conducted regional-based rainfall forecasting for the Ratnapura district and predicted the effect of rainfall on disasters, agriculture, and hydropower generation.

2. Background and Literature Survey

1. Regional-wise rainfall prediction for Sri Lanka

Rainfall is one of the most important weather conditions in Sri Lanka. Forecasting possible rainfall can help to solve several problems related to the tourism industry, natural disaster management, agricultural industry, etc. As the Sri Lankan rural economy is mostly based on agriculture, it is important to forecast rainfall as well as other weather conditions accurately. There have been several studies regarding Machine Learning Approaches for Regional wise rainfall prediction in Sri Lanka. But very few of them have been regarding the Ratnapura district. Some of them are mentioned below with their conclusions. In recent years, there has been a growing interest in the use of advanced machine learning algorithms for rainfall prediction. Machine learning algorithms can learn from historical data to identify patterns and relationships, which can then predict future rainfall events.

There are numerous existing rainfall prediction approaches proposed by researchers through their studies about statistical models and data analytic techniques for predicting future weather in terms of different weather-related variables. Data mining techniques such as regression, decision trees, clustering, neural networks, and many others are being used to identify the most accurate and efficient techniques for predicting weather based on statistical models [10]. [11] Has provided a rainfall prediction model that improves accuracy by combining data mining and machine learning methods. Their research found that ARIMA models and neural networks provided the highest levels of accuracy. [11] developed a neural network model for Sri Lanka's three regions, predicting rainfall with high accuracy using historical, meteorological, and topographic data. [12] Demonstrates that the SARIMA model is considered by most studies to be the best conventional statistical model for the time series forecasting of rainfall. However, modern data mining techniques are increasingly being used by researchers in place of conventional statistical models. Artificial neural network (ANN) models, one of the more recent data mining technologies, outperformed more conventional models in terms of prediction quality. A rainfall prediction model developed by [13] A. S. &. F. S. M. Jayasekara, based on Artificial Neural Networks, is an empirical method-based prediction approach. The number of hidden layer neurons needed for the model must be calculated in these types of approaches because the time needed for model training excessively increases with the number of neurons. It offers a method for resolving a variety of nonlinear issues that are challenging to resolve using conventional methods. As a result of these studies, Sri Lankan rainfall prediction models can be developed using advanced machine learning algorithms. It is, however, necessary to conduct more research in order to develop models capable of predicting rainfall at finer spatial and temporal scales. A further need for research is to develop methods for incorporating climate change scenarios into rainfall prediction models.

2. Regional wise Natural Disaster (Flood)Forecasting

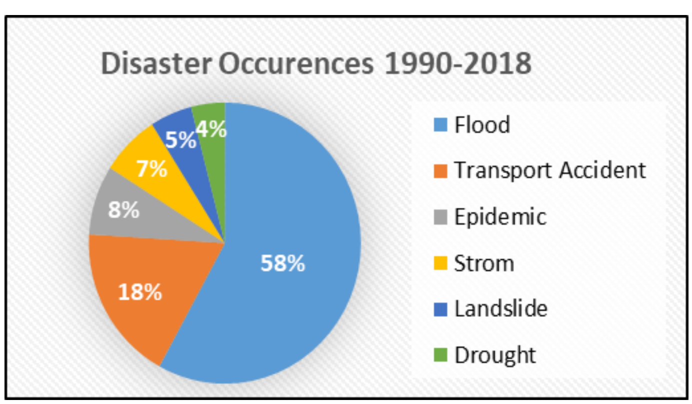


Figure 1.1: Disaster Occurrences 1990-2018

According to the Ministry of Disaster Management Center in Sri Lanka, floods are the most recurrent natural disaster in Sri Lanka, leading to significant harm to human lives, infrastructure, agriculture, and the economy. As well as being used for forecasting, machine learning techniques have also been explored for assessing disaster risk and analyzing vulnerabilities. As an example, researchers have employed machine learning to estimate the human and economic losses from disasters and to assess the vulnerability of buildings to natural hazards. Integration with technologies like remote sensing and GIS has further improved accuracy in disaster forecasting. To disseminate crucial information effectively, there must be more localised and precise prediction models integrated into warning systems.

There have been many successful applications of machine learning models to flood prediction. Youssef et al. [4] Utilized support vector machines (SVM) to predict floods in Turkey's Denizli Basin using meteorological data. In another investigation, W. A. M. Prabuddhi and B. L. D. Seneviratne [3] employed a Recurrent Neural Network model for the flood prediction in Deduru Oya, Sri Lanka. Another study based on an extreme learning machine, an artificial neural network (ANN), for flood prediction in India's Mahanadi River Basin surpassing other models like SVM and random forests.

Gamage et al. [5] found that comparisons between traditional regression models and machine learning models for flood prediction in Sri Lanka revealed the superiority of machine learning, particularly artificial neural networks (ANNs), Karunarathna et al. [6] developed a hybrid model combining wavelet transform and machine learning algorithms that demonstrated high accuracy in flood forecasting for the Kelani River basin.

Further studies employed by Ahmad et al. [7] and Lim et al. [8] in flood forecasting for the Indus River basin and Malaysia's Klang River basin, highlighted the enhanced performance of these models over traditional statistical methods. The various machine learning approaches, including SVM, ANNs, and hybrid models, have proven effective in leveraging input variables like precipitation, temperature, humidity, and evapotranspiration to accurately predict flood events. In summary, machine learning has emerged as a powerful tool in flood prediction, offering advanced models that outperform traditional methods and enhance disaster preparedness and response.

3. Paddy harvesting prediction

The lovely island nation of Sri Lanka's Rathnapura District is known for its fertile plains and verdant surroundings. The district's economy is based primarily on agriculture, with paddy cultivation being one of the main agricultural pursuits. Rice, commonly known as paddy, is a key crop in Sri Lanka, and a number of variables, including rainfall, affect its productivity. Rathnapura District's agricultural environment is significantly shaped by rainfall. A distinct rainy season and dry season are present in the district's tropical rainforest climate. The success of paddy agriculture is directly impacted by the quantity and distribution of rainfall. Heavy monsoonal rains during the wet season, which normally lasts from May to September, give paddy fields the moisture they require. When there is sufficient rainfall during this time, paddy plants receive the water they need for growth and development. On the other hand, a lack of rainfall during this crucial stage can result in lower yields, which will have an impact on the livelihoods of several farmers in the area. Conversely, irrigation is necessary to maintain paddy crops during the dry season, which lasts from December to February. To effectively harness and manage water supplies, farmers in Rathnapura District have created sophisticated irrigation systems, including antiquated cascading tank systems. In the past, these systems have proven crucial for reducing the effects of unpredictable rainfall patterns. Recent years have seen increased difficulties with paddy production in Rathnapura District due to climatic change. Crop yields are seriously threatened by unpredictable weather patterns, especially erratic and intense rainfall events. To lessen these risks, many farmers are implementing climate-resilient agricultural practices and diversifying their crop production. In conclusion, rainfall patterns in the area play a significant role in predicting paddy yield in the Rathnapura District. The thorough monitoring and analysis of rainfall, 14 which is the lifeblood of agriculture in this lush region, is crucial for ensuring food security and maintaining the livelihoods of the surrounding farming community. Innovative methods for predicting rainfall and adaptable farming techniques will be essential for the future of paddy production in Rathnapura District as climate change continues to impact weather patterns. ANN Structure is derived from biological neural processes in the human brain. The technique is developed using the relationships between the neurons. Several studies for crop prediction have discussed the potential of ANN use. A study forecasted rice yield using climatic predictions figures at most 40-60 kg per hectare. [2] One more by [3] has been cited as describing a straightforward and accurate tool for estimating rice production. Rice production forecast using the application of the ANN model was also examined for South Asia. During training, ANN provides acceptable errors. That describes how accurate a model is. Next, in a comparison of the output, the ANN result reveals the same result that validates the accuracy of the forecast. In the Siraha district of Nepal, the anticipated effect might be applied to increase paddy yield [4]. Other recent studies have shown the effectiveness of a DSS in Thailand's Phimai district that employed ANN to forecast rice production [5]. Details from another learning that proved the use of ANNs with feed-forward back propagation for farming produce forecasting were related results

4. Hydropower generation Prediction

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.

3. Research Gap

The Rathnapura district of Sri Lanka is prone to high rainfall variability, which has a significant impact on agriculture, water resources, and flood control. However, there is a lack of advanced machine learning-based rainfall prediction models that are specifically tailored to the district. Existing models often rely on regional data, which may not adequately capture the unique geographical and meteorological characteristics of Rathnapura. The topographical diversity of Rathnapura, with its mountainous terrain, and dense forests creates complex microclimates and localized weather patterns. This requires a specialized approach to rainfall prediction. Additionally, Rathnapura's agricultural economy and vulnerability to floods highlight the need for precise and highly localized rainfall predictions. Agriculture is culturally and economically important in Rathnapura, so models that can provide accurate forecasts are essential to support planning and mitigation efforts for farmers and local authorities.

This study covers a significant research gap in the development and implementation of an integrated approach that combines the floods, agriculture, and hydraulic power generation predictions with regional wise rainfall prediction regarding to the Ratnapura district. There are 4 separate machine models developed in the system while the rainfall prediction model works as the centralized model. The main prediction of this system is regional wise rainfall prediction and others (predictions of floods, agriculture, and hydraulic power generation) are predicted on the results of the rainfall prediction model.

3.1 RESEARCH PROBLEM

1. Overall research problem

"How can accurately predicting the continuous rainfall in Ratnapura district during the relevant periods and Provision of rainfall data required to make predictions in the fields of flood disasters, agriculture and hydropower generation using advanced machine learning techniques?"

The main research problem of this study is to Accurate forecasting of the continuous heavy rainfall in Ratnapura district from the south-west monsoon which lasts from May to September using advanced machine learning techniques.so that it is possible to make the prediction about the impact of rainfall on flood, agriculture, and hydropower generation. In addition to that accurately forecasting the daily, weekly, and monthly rainfall of Ratnapura district regarding the rest of the durations of the year.

Here the importance of the research problem is due the Rathnapura district faces significant flooding risks during monsoon season, necessitating accurate rainfall predictions for disaster preparedness and response to safeguard lives and property, and in Ratnapura the agricultural sector relies heavily on monsoon rains, requiring accurate rainfall forecasts for informed crop planting, irrigation, and harvest planning, as incorrect predictions can be financially damaging. Furthermore, hydropower generation in Rathnapura is intricately linked to the south-west monsoon's rainfall as well as an Accurate forecasting is crucial for optimizing hydropower production and ensuring stable energy supply, especially in a world embracing sustainable and renewable energy sources.

Even though the main purpose of this research problem is to develop accurate and reliable forecasting models for continuous heavy rainfall in the Rathnapura district, during the south-west monsoon seasons. It serves some additional purposes such as mitigate the impact of flooding with developing early warning systems, supporting the agriculture with allowing farmers to make informed decisions about when to plant, irrigate, and harvest crops and optimizing the hydro power generation and ensuring the stable energy supply.so above it clearly describes the purposes and the importance of the research problem.

3.1.1 Research Problem addressed by regional-wise rainfall prediction system for Ratnapura district.

Sri Lanka mainly experiences two main monsoon seasons called the Southwest Monsoon and the Northeast Monsoon. Rathnapura district takes a prominent place among the areas that receive rain from southwest monsoon. Thus, due to the inability to predict the heavy rainfall in the Ratnapura district in advance, many cases of natural disasters such as floods and the destruction of agricultural crops have been reported. the Ratnapura district in the Sabaragamuwa province of Sri Lanka has been affected by floods due to rainfall, agricultural crops have been damaged due to adverse weather extremes, and has the potential to collect water for hydropower generation. For these reasons, this study conducted regional-based rainfall forecasting for the Ratnapura district and predicted the effect of rainfall on disasters, agriculture, and hydropower generation.

The main purpose of this research problem is to develop accurate and reliable forecasting models for continuous heavy rainfall in the Rathnapura district, during the southwest monsoon seasons as well as the whole the year as the long-term and long-term prediction.

3.1.2 Research Problem addressed by natural disaster forecasting system.

As one of the districts within Sri Lanka that has received the highest amount of annual rainfall is the Rathnapura district. Consequently, due to the inability to predict the heavy rainfall in the Rathnapura district in advance, many cases of natural disasters have been reported such as floods and the destruction of agricultural crops in the Rathnapura district as a result of the inability to predict the heavy rainfall. In the Sabaragamuwa province of Sri Lanka, flooding has been caused by heavy rainfall, agricultural crops have been damaged by adverse weather conditions, and the potential exists to collect water to generate hydropower in the area. Because of these reasons, this study performed regional-based rainfall forecasting for the Rathnapura district and predicted the effects of rainfall on disasters, agriculture, and hydroelectric generation based on regionally-based rainfall forecasting.

This research problem aims to address the pressing challenges that are being faced by the Rathnapura district, which is primarily influenced by the unique climatic conditions and geographical characteristics of the district. The core research problem is defined as follows.

Enhancing Rainfall Prediction and Impact Assessment

The study's goal is to create a sophisticated machine learning-based rainfall prediction system tailored specifically to the Rathnapura district. This system will be focused on providing accurate, localized rainfall forecasts while taking into account the district's unique meteorological characteristics. By doing so, it hopes to improve the region's ability to predict heavy rainfall events, resulting in more effective disaster management, particularly in the context of floods.

Precise Rainfall Prediction

The goal of the study is to develop a precise rainfall prediction model that considers the distinctive topography, proximity to water bodies, and other geographical aspects of Rathnapura. The objective is to provide precise and timely rainfall forecasts so that communities and authorities can plan ahead and proactively respond to potential flood events.

Managing Agricultural Vulnerability

Since agriculture makes up a large portion of Rathnapura's economy, it is vulnerable to the effects of heavy rainfall, including crop damage and soil erosion. By offering timely rainfall forecasts that can help farmers make knowledgeable decisions about planting, harvesting, and soil management, the research aims to address this vulnerability and minimize agricultural losses.

Harnessing Hydropower Potential

The district's abundant rainfall resources present an opportunity for long-term hydropower generation. In order to manage water resources for hydropower as efficiently and dependably as possible, the research aims to create models for predicting rainfall.

3.1.3 Research Problem addressed by paddy harvesting prediction system

A crucial and complex issue, the research subject of estimating rice production using rainfall data in the Rathnapura District has substantial consequences for Sri Lankan agriculture, food security, and rural lives. Accurate yield forecasts are crucial because this area, well known for its significant contribution to the nation's rice production, depends on the monsoon rains. With better resource allocation, risk reduction, and overall agricultural sustainability possible, this research challenge aims to address the urgent need for a trustworthy and locally customized prediction model that can inform farmers, regulators, and stakeholders about prospective crop outcomes. The Rathnapura District experiences different wet and dry seasons due to its location in a tropical monsoon climate zone. The volume, distribution, and timing of rainfall directly affect the soil moisture levels necessary for rice growth in this region, which are directly correlated with the success of paddy agriculture. Therefore, the focus of this research is on creating a reliable predictive model that can use historical rainfall data to reliably estimate paddy yield results. The great variation in rainfall patterns within the Rathnapura District is one of the most noticeable difficulties posed by this study subject. The area has microclimates that, 19 even within exceedingly small geographic areas, can result in significant differences in the distribution of rainfall. Therefore, to develop a trustworthy predictive model, it is essential to gather and examine high-resolution rainfall data at the sub-district or even village level. Additionally, to improve the model's precision and thoroughness, additional meteorological elements like temperature, humidity, and wind patterns may need to be considered. To build the prediction model, the proposed research would make use of innovative machine learning and data science methodologies. This model will be built using painstakingly matched historical rainfall data over multiple decades and paddy yield records. Regression analysis, neural networks, and ensemble approaches are just a few examples of the machine learning algorithms that will be used to create a solid, data driven relationship between rainfall and paddy output. To guarantee the model's dependability and generalizability, it will undergo rigorous training and validation using cross-validation approaches. Additionally, this research issue acknowledges how the Rathnapura District's rainfall patterns may soon be impacted by climate change. Understanding how global climate change affects paddy production is becoming more important as precipitation patterns continue to change. To make the predictive model more adaptable and produce more precise long-term predictions, the research will investigate creative ways to incorporate climate change projections into it. The findings of this project are expected to have important practical ramifications. The Rathnapura District's farmers stand to gain a great deal from accurate and timely production estimates. With this knowledge, they are better equipped to choose crops, arrange planting times, and allocate resources, thus improving their chances of farming success. The model's insights can also be used by politicians and agricultural authorities to establish well-informed plans for disaster preparedness, climate adaptation, and agricultural development—all of which are essential for the region's sustainable growth and food security. Using rainfall data to estimate paddy output in the Rathnapura District is a complex research subject that necessitates a multidisciplinary approach. This study intends to provide important insights into the complex interaction between rainfall patterns and agricultural outcomes by leveraging the capabilities of data science, machine learning, and climate modeling. In the end, the creation of a precise predictive model will boost the resilience and sustainability of the rice industry in this crucial agricultural region, ensuring Sri Lanka's economic stability and food security

3.1.4. Research Problem addressed by hydropower generation system.

"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?"

This research problem facilitates the transition from conventional energy reliance to the harmonious coexistence of hydropower and alternative energy sources. 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 and 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.

This integrated approach's successful implementation can 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.

Due to this research problem, the ebb and flow of nature can be seamlessly incorporated into energy production, minimizing disruptions and safeguarding the environment.

3.2 OBJECTIVES

The Main Objective

The main objective of this research is to develop an integrated machine learning-based framework for regional-based rainfall prediction and predicting impact of rainfall on flood, agriculture, and hydropower generation for Ratnapura districts using combined machine learning models.

Specific Objectives

Develop a regional-specific rainfall prediction model for the Ratnapura district to do both short- and long-term rainfall prediction with better accuracy.

Identify and evaluate different machine learning models for regional-wise rainfall prediction.

Develop a region-specific flood prediction model for the Rathnapura district, with a particular focus on the Kalu Ganga River basin, by leveraging data-driven insights from heavy rainfall patterns.

Identify and evaluate different machine learning models for region-specific flood prediction.

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.

Evaluate and select appropriate machine learning algorithms for power consumption prediction.

Analyze power consumption patterns in the target area, considering the complex relationships between weather conditions, consumer behavior, and other relevant factors.

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.

Develop and validate a predictive model using advanced machine learning techniques to forecast the paddy yield of Rathnapura district based on rainfall patterns.

4. METHODOLOGY

Methodology

This section emphasizes the four key components each interconnected as depicted in Fig.1 below. In the following subsections, the study provides more information on the study strategy, research technique, research methodology, data collection methods, sample selection, research procedure, and limitations of the project research.

The four major components of this proposed Web application are listed below.

Rainfall Prediction

Flood Forecasting

Paddy yield prediction using precipitation

hydropower generation

4.1 Regional-wise rainfall prediction model development

past Research Analysis

Several studies have been conducted using machine learning techniques to forecast rainfall in Sri Lanka's Ratnapura district. ML models can learn complex patterns in historical rainfall data and use this knowledge to make predictions about future rainfall events. So that there has been several ML based prediction developed to forecast the rainfall in Sri Lanka. here are some examples of past research on regional rainfall prediction using ML techniques.

1. T. Dananjali, S. Wijesinghe and J. Ekanayake, "Forecasting Weekly Rainfall Using Data Mining Technologies,". The SVM-FA model outperformed other models, including a single SVM model and a multiple linear regression model.
2. M. Gunawardana, H. Amarasekara and M. Perera, "Short-term precipitation prediction in Sri Lanka using support vector machines," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.The RF-SVM model outperformed other models, including a single RF model and a single SVM model.
3. Y. Tikhamarine, D. Gamane, A. Ahmed, O. Kisi and L. Shafie, "Improving artificial intelligence models accuracy for monthly streamflow forecasting using grey Wolf optimization (GWO) algorithm," in J. Hydrol, 2019.

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.

Identifying Existing Systems

Exactly Several systems use machine learning techniques to predict regional rainfall. Some of these systems are available to the public. Numerous characteristics are commonly shared by existing machine learning-based regional rainfall forecast systems, included.

Use of historical rainfall data: Training data is necessary for all machine learning models to become intelligent. This training data is usually historical rainfall data from the location where the model will be used for rainfall prediction.

Use of additional meteorological data: Machine learning models for rainfall prediction frequently make use of additional meteorological data, such as temperature, humidity, and wind speed, in addition to historical rainfall data. The models may be able to understand more intricate correlations between various climatic factors and rainfall with the use of this extra data.

Several machine learning methods have been utilized: Support vector machines (SVMs), random forests (RFs), neural networks (NNs), and gradient boosting machines (GBMs) are among the machine learning algorithms that have been used to predict regional rainfall. The ideal method for a given application will rely on the desired model performance as well as the unique characteristics of the data.

Even with the advancements in recent years, there are still certain shortcomings with the current machine learning-based regional rainfall forecast systems. Several of these flaws consist of them.

Sensitivity to the quality and quantity of the training data: Machine learning models are only as good as the data they are trained on. If the training data is poor quality or incomplete, the model will not be able to learn accurate relationships between the meteorological variables and rainfall.

Limited ability to predict rainfall at finer temporal and spatial scales: Most machine learning algorithms in use today are not able to generate forecasts at incredibly fine temporal and spatial scales. This is because high-resolution input data, which is frequently unavailable, is needed for such predictions to be made.

Proposed Architecture

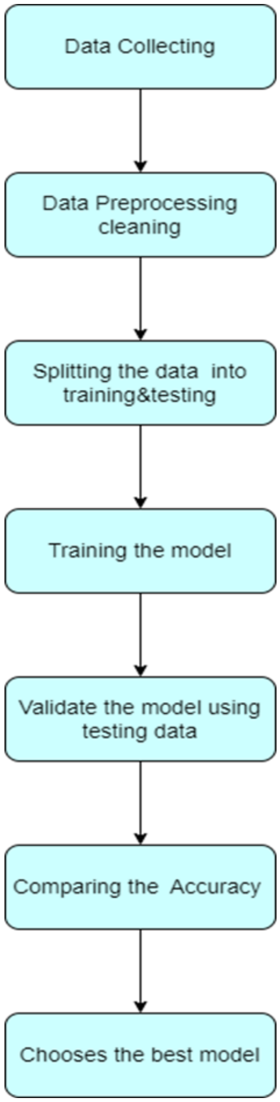


Figure 2 Proposed architecture

System Development and Implementation

The development and implementation of our integrated machine learning-based rainfall prediction system for the Rathnapura district represents a technically robust endeavor. Our meticulous model selection process led us to favor the Long Short-Term Memory (LSTM) model over competing methodologies such as Auto-Arima and neural prophet, primarily based on its superior performance in minimizing root mean squared error, thereby ensuring the highest standards of precision and reliability in our precipitation forecasts. 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.

In constructing the system's architecture, Python Flask has been strategically employed for the backend, capitalizing on its lightweight, efficient, and versatile nature to handle the data processing and model interactions seamlessly. The frontend is meticulously crafted using React, providing users with an intuitive and responsive interface to access and engage with real-time predictions. For efficient data storage, retrieval, and management, we've harnessed the power of a MongoDB cloud database. Its NoSQL architecture accommodates the vast dataset requirements and facilitates horizontal scaling as the data volume expands. The connection between the frontend and MongoDB cloud database is realized through meticulously designed APIs, enabling users to access the most recent rainfall forecasts in real-time.

This system has been designed with scalability and adaptability in mind, not just to meet the unique needs of Rathnapura but also to allow for future expansion to serve other areas. Our project is a compelling example of how advanced predictive technologies can contribute to meteorological advancements, disaster mitigation, and data-driven insights for a multitude of industries worldwide. It does this by combining state-of-the-art machine learning techniques with a robust technical infrastructure.

Rainfall Prediction Model

The proposed methodology consists of several steps including acquiring weather observation data, data preprocessing, designing a machine learning model, training a machine learning model, making predictions on the test set, and finally evaluating the predictions made by the model.

a. Data Collection

Weather observation data is collected from the http://meteomanz.com/ website, which is a worldwide wild online metrological data center, the data is collected starting from 2003/01/01 to 2023/01/01 for Rathnapura weather station of Sri Lanka. The data include daily weekly, and monthly rainfall data regarding the Rathnapura district.

b. Data Preprocessing and Cleaning

A methodical strategy was used to guarantee the integrity and quality of the dataset during the data preparation and cleaning stage, which was necessary for training the machine learning model of rainfall prediction. To reduce the focus on crucial information, the dataset was first carefully trimmed to contain only the date and precipitation columns. A dependable and continuous series of precipitation data was established for analysis, with a consistent date range of September 30, 2009, to May 30, 2023. 'Tr' records were replaced with the value 0.0 as part of additional data refining, which resolved a common notation problem. Null values were substituted for '-' entries, and a thorough investigation was conducted to find any missing dates by generating a date range and comparing it to the date list in the dataset. With linear interpolation, missing data points were meticulously addressed to ensure completeness and accuracy, resulting in a robust, comprehensive dataset that served as a basis for training a machine learning model. As a result of these preprocessing and cleaning techniques, a high-quality dataset was generated, which set the stage for accurate and reliable rainfall predictions.

c. Model Implementation

Model is trained using three machine learning algorithms:

I. Auto Arima
II. Neural Prophet
III. LSTM

The LSTM algorithms is chosen as the best algorithm fit for the model. The main reason to choose here the LSTM model is well supported to the long sequential data processing and with higher accuracy and lower RMSE loss to make a prediction using them.

The machine learning model is built with the LSTM (Long Short-Term Memory) architecture, which is well suited for handling sequential data. During training, the model's learning rate of 0.0003 ensures gradual and stable convergence. An embedding layer with a dimension of 16 improves the representation of input data, and the LSTM component is built with 32 units to capture complex temporal patterns. A batch size of 16 is used to train the model, allowing for parallel data processing.

d. 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.

e. 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.

f. 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.

g. 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 rainfall 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 regional wise rainfall prediction using an LSTM algorithm. It also uses a custom callback to save the best model during training in order to ensure peak performance.

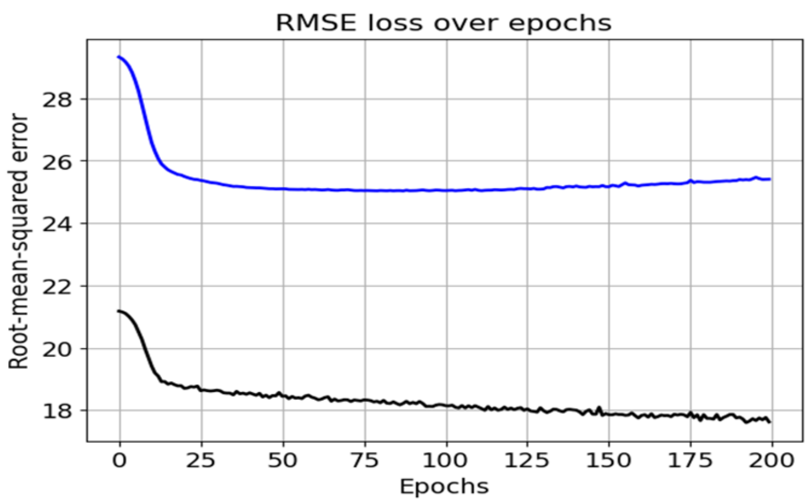


Figure 3 RMSE of the prediction mode

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

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

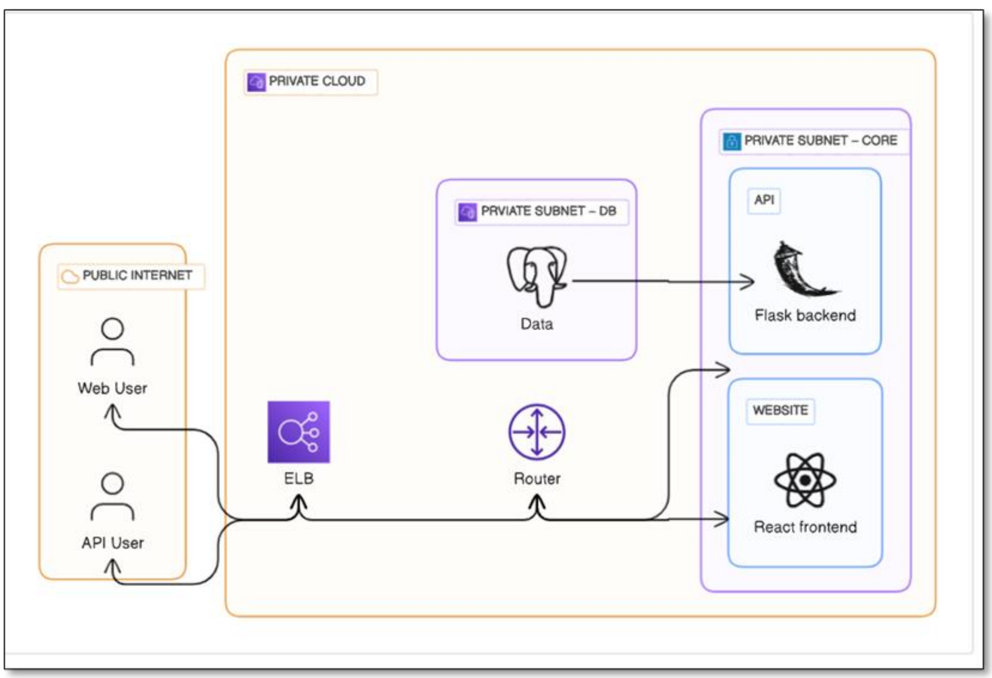


Figure 4 Tool and technology

Design of web applications

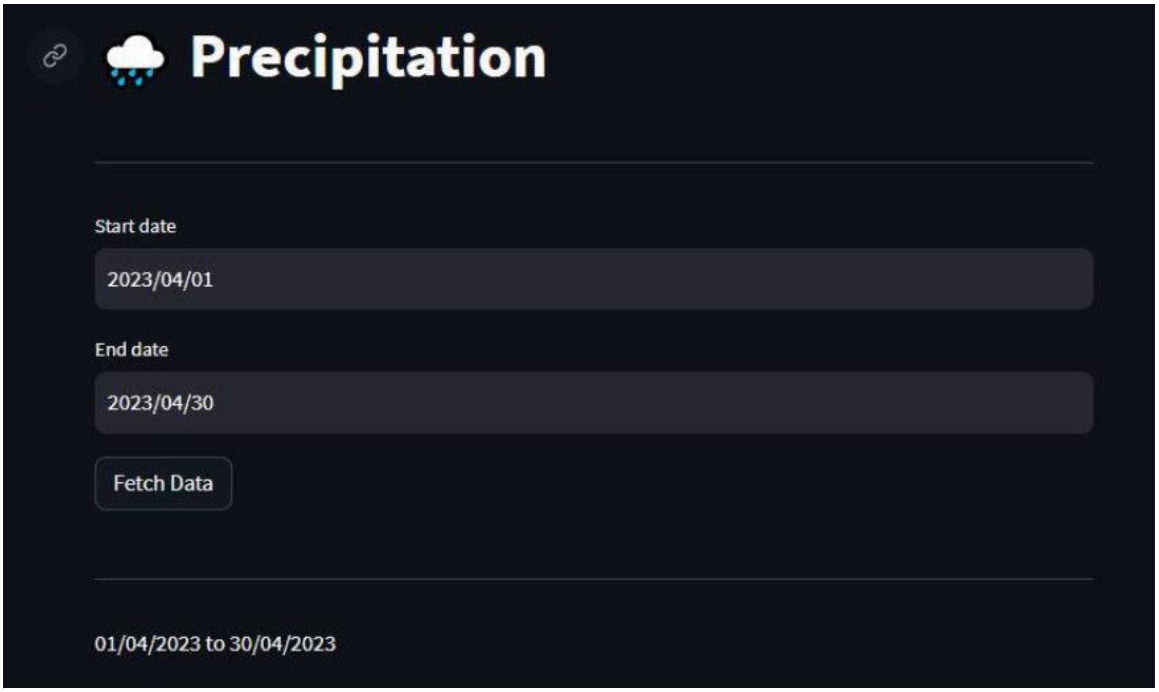


Figure 5 Frontend UI-1

Figure 6 Frontend UI-2

4. 2. Flood forecasting model development

Requirement Gathering

An essential step that serves as the foundation for the research is requirement gathering. It entails a thorough investigation of the body of information and sources already available that are pertinent to the goals of the study. The details of requirement gathering are covered in the ensuing subsections.

Past Research Analysis

This stage involves a thorough examination of prior research, with an emphasis on regional weather forecasting. This analysis identifies critical insights into the state of the field today, prevalent methodologies, and knowledge gaps where additional research is necessary. This analysis provides a helpful framework for defining the research's approach and goals by analyzing the benefits and shortcomings of earlier research.

Identifying Existing Systems

The research seeks and evaluates current systems, tools, and software programs designed for rainfall prediction using historical rainfall data in order to build a robust framework for rainfall forecasting. This necessitates a thorough examination of research projects, software applications, and agricultural decision support systems with predictive capabilities. Finding and studying these existing resources will help with the selection of methodologies and technologies for this research and will also enable a more informed and efficient approach to regional-based flood prediction.

Feasibility Study

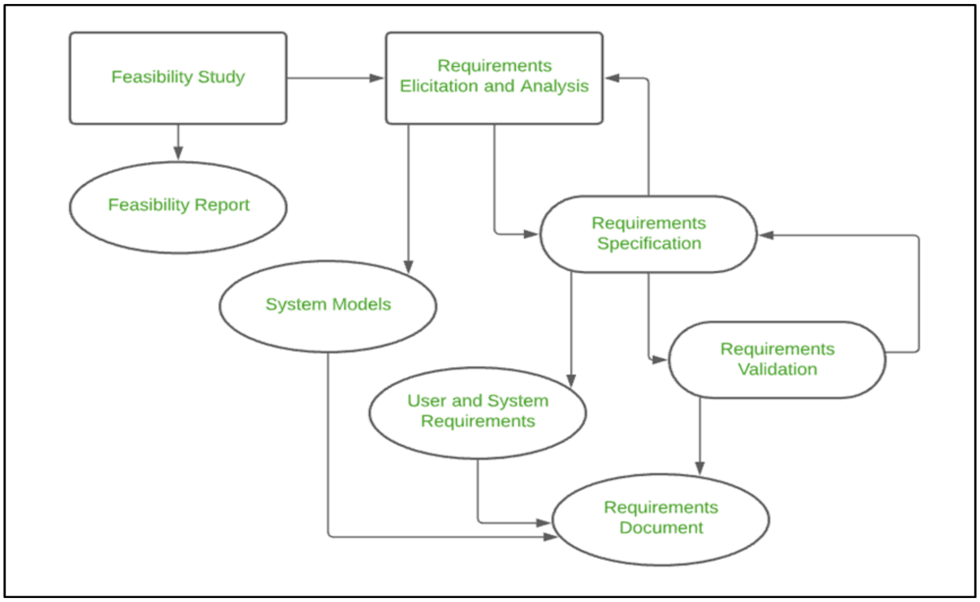


Figure 4.2: Feasibility Study Analysis

Technical Feasibility

The technical feasibility of a research project is determined by its ability to effectively use advanced machine learning techniques, data processing, and computational resources. In this context, the research project takes advantage of advanced machine learning libraries and tools, as well as high-performance computing infrastructure for data analysis and model development. The technical feasibility is supported by the existence of well-established frameworks and methodologies for rainfall prediction, which provide a solid foundation for the research.

Schedule Feasibility

The ability of the project to adhere to a predefined timeline is assessed by schedule feasibility. Given the complexity of developing accurate rainfall prediction models and conducting comprehensive data analysis, a reasonable timeline has been established. The study is designed to accommodate data collection, preprocessing, model development, and evaluation within the time frame specified. Effective time management strategies and project management tools will be used to ensure that the project stays on track and meets its milestones on time.

Requirement Analysis

The requirement analysis phase is essential because it entails determining the precise requirements and needs of the study. The following essential components are included in this phase.

Data Requirements

To determine the data's availability, quality, and relevance, a thorough examination of data sources such as meteorological records, river gauge data, and historical flood datasets is performed. This analysis helps in choosing the data sources that are essential for developing and testing the predictive models.

Model Requirements

The research project describes the specific machine learning techniques and algorithms required for accurate rainfall prediction. It also specifies the computational and software tools required for model development and evaluation.

Personnel Requirements

The roles and specialities of the research team members, including data scientists, machine learning specialists, subject matter experts, and project managers, are described. This evaluation confirms that the project has the necessary personnel to carry out the research efficiently.

Infrastructure Requirements

To make sure it meets the technical requirements of the research, the computational infrastructure, including hardware and software, is evaluated. This analysis aids in determining whether any infrastructure upgrades or improvements are necessary.

System Analysis

A thorough system analysis is required for the complicated and interdisciplinary task of predicting the rainfall for Ratnapura weather station daily, weekly, and monthly using past precipitation data. An efficient rainfall forecast system is designed, developed, and implemented based on such an analysis. This procedure entails a thorough analysis of the system's numerous parts, interactions, and requirements.

Gaining a comprehensive grasp of the project's scope, its technical and operational components, and the needs of the associated stakeholders is the main goal of system analysis. System analysis offers the foundation for wise decision-making, resource allocation, and the creation of a strong predictive system by outlining these essential components.

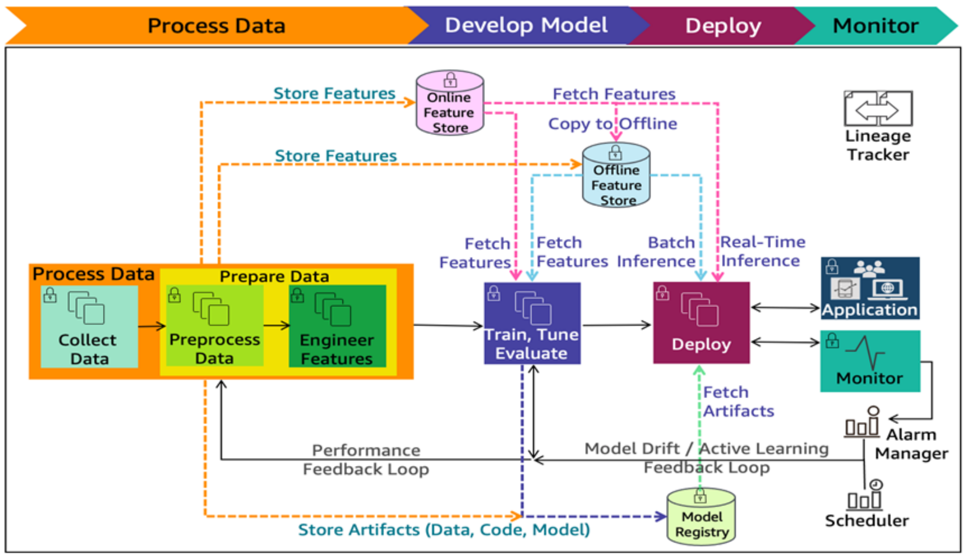


Figure 4.4: System Architecture

a. Data Collection

The data sets used in this study were collected from various sources, including the Department of Disaster Recovery web page, online freely available data centers and satellite observations. The dataset covers a period of 13 years (2010 - 2023) for a specific region based on Kalu Ganga river basin in Sri Lanka. The dataset used in this study includes historical rainfall data and corresponding flood statuses (Normal, Alert, and Flood).

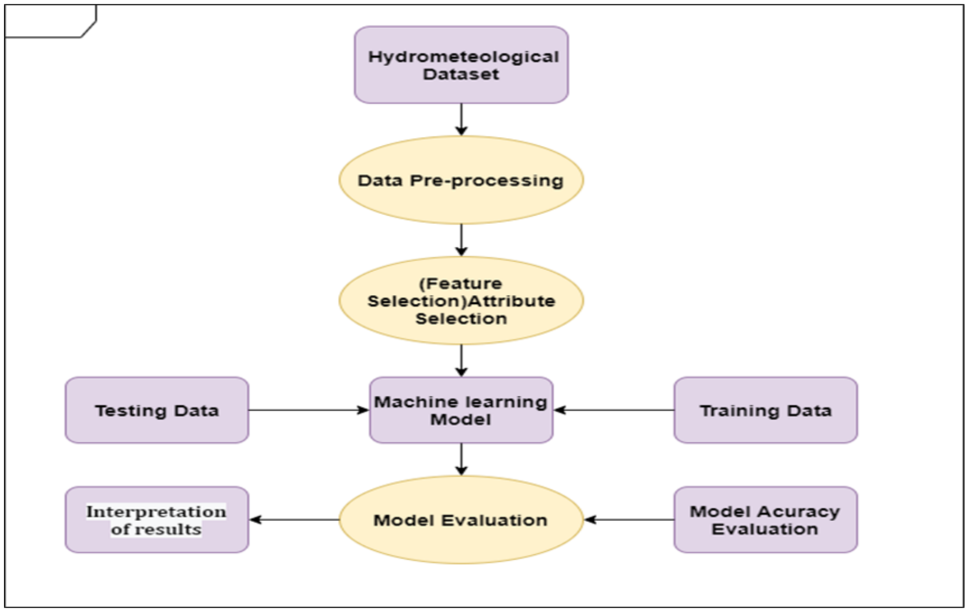


Figure 4.5: Methodological Diagram

b. *Data Preprocessing and Cleaning.*

Data pre-processing plays a vital role in preparing a dataset before model training, serving as a fundamental step. It involves several key procedures such as data cleaning, data encoding, and train set splitting. In order to mitigate the occurrence of overfitting, it was essential to partition the dataset into distinct training and testing sets. The data is then divided into train and test sets, where 75% is allocated for training and the remaining 25% for testing. Class balancing techniques are applied to address the imbalanced distribution of flood statuses. Additionally, it is worth noting that the size of the datasets, along with the ratios employed for the train/test split, can exert a substantial influence on the output generated by the model. These factors can directly impact the overall classification performance.

i. Handling Missing Data

- Missing values in the "rainfall" and "Flood_Status" columns were handled. Rainfall data was interpolated using a linear method, while missing flood status values were filled using forward fill. If the missing value rows only make up a small fraction of our dataset and won't have a big impact on the study, we removed them.

ii. Handling Duplicates

- Checked for duplicate entries in our dataset and removed them to ensure data integrity.

iii. Data Validation

- Cross-checked our data with trusted sources to validate its accuracy. This can involve comparing our data with data from meteorological agencies or government sources.

iv. Feature Selection

- Correlation analysis was performed to identify the features that have the highest correlation with the target variable (flood occurrences). Additionally, domain experts were consulted to select important variables based on their knowledge of flood dynamics in the region.

v. Feature Engineering

- Lagged variables were created by incorporating the values of the target variable and other relevant predictors at previous time steps. Statistical aggregates such as mean, median, and standard deviation were calculated for selected variables over different time windows (e.g., 7 days, 30 days) to capture temporal patterns and trends.

System Development and Implementation

Data Virtualization

Data visualization is an essential first step in our system development before we proceed to the model implementation. It enables us to discover patterns in the dataset, understand it visually, and come to wise decisions. The historical rainfall data from a specific time period—January 1, 2011, to January 1, 2012—will be visualized in this section. Prior to beginning the data virtualization process, we must make sure that our dataset is properly organized. This section will set the stage for our virtualization efforts by briefly describing the steps in data preparation.

```
# Data Preparation
df_final = merge1.set_index('Date')
# Filter Data
mask_1 = (df_selection['Date'] >= '2011-01-01') & (df_selection['Date'] <= '2012-01-01')
```

```
df_temp = df_selection.loc[mask_1].sort_values('Date')
df_temp = df_temp.set_index('Date')
```

Now that we have our data ready, let us begin the data virtualization process. We will create a time series plot that depicts rainfall trends over the specified time period (January 1, 2011 to January 1, 2012).

Figure 4.5.1: Rainfall Data Virtualization

The plot above shows a clear representation of rainfall patterns over the specified time period. This virtualization provides a visual understanding of how rainfall fluctuates during this critical period, identifying temporal patterns, evaluating data quality, and comparing historical data. It is an essential step in our flood prediction system because it enables us to find patterns and trends that could affect flood events.

Model Development.

Before deciding on the KNN model for our dataset, we perform training and testing using another model, LSTM as well. After evaluating the performance of each model, we select the KNN model as it demonstrates the highest accuracy (96%). The KNN algorithm was also selected for flood prediction due to its efficiency and simplicity. KNN is a supervised learning algorithm used for classification tasks. It works by locating the k-nearest data points to a given data point in the training set and classifying it based on the majority class among its neighbors.

The K-Nearest Neighbors (KNN) algorithm is the foundation of our flood prediction model, which uses historical rainfall data to forecast flood events. The model determines whether a future scenario corresponds to a "Flood," "Alert," or "Normal" condition by considering the nearest historical data points with similar rainfall patterns. A series of procedures were used to create the model, including data encoding and decoding, data conversion, class balancing, train-test splitting, KNN model training, model evaluation, and inference. All the Python code references corresponding to the various model implementation sections are provided in a separate section cited in the appendix below.

Custom Data Encoding and Conversion

This step is essential because the KNN model functions by identifying the training set data points that are most similar to a new data point. The data must be organized in a way that makes it simple to calculate the distances between data points in order to accomplish this. In order to transform the data into a matrix format, we first create a list of lists. The outer list represents all of the data points in the dataset, and each inner list represents a single data point. The inner list then contains each data point's features in the order in which they were originally collected.

The target variable "Flood_Status" is transformed into one-hot encoded vectors using a custom encoding function you have created called "custom_encoder." "Flood" is encoded as [1, 0, 0], "Alert" as [0, 1, 0], and "Normal" as [0, 0, 1]. With the help of the convertToMatrix function, you can model your raw data. By using a series of rainfall data as the input and the corresponding encoded flood status as the output, it creates input-output pairs.

Class Balancing.

The KNN model is a supervised learning algorithm, which means it must be trained on a dataset with labeled data. When it comes to flood forecasting, the labels would indicate the status of each data point with regard to flooding (for example, flood, alert, or normal). It is important to note, however, that the label distribution in the dataset may not be balanced. For instance, "normal" data points might be much more prevalent than "flood" or "alert" data points. The KNN model may then be biased towards predicting the "normal" class as a result.

To address the class imbalance in the dataset, I have implemented a function called balance_df. It divides data points into lists based on their class, oversamples each class to the size of the largest class, and then shuffles the data to create a balanced dataset.

Data Normalization

It is necessary to normalize the data before training the KNN model. This entails scaling the data so that all features are on the same scale. This is required because the KNN model employs distance calculations to identify the data points that are most similar to a new data point. The distance calculations may be biased in favor of features with higher values if the features are not scaled on the same plane.

The 'norm' function is used to normalize the data you enter. It ensures that each input vector sums to one, making the model more resistant to variations in data scale.

Train and test data Splitting

We must divide the data into training and test sets after it has been balanced and normalized. The KNN model will be trained using the training set, and its performance will be assessed using the test set. As a general rule, divide the data into a training set and a test set in the proportion of 80:20. Accordingly, 80% of the data will be used for training and the remaining 20% for testing.

Using train_test_split from the sklearn library, I have divided the balanced dataset into a training set and a test set. For reference, I have printed out the shapes of these datasets. Following that, the pickle library was used to save the training and test datasets for later use.

KNN model Fitting.

We use the fit() method of the NearestNeighbors() class from scikit-learn to fit the KNN model to the training data. The number of nearest neighbors to take into account when predicting the class of a new data point is specified by the n_neighbors parameter. The KNN model will be more noise-resistant with a higher value of n_neighbors, but it will also be more likely to overfit the training set of data. A smaller value of n_neighbors will increase the KNN model's noise sensitivity while lowering the likelihood that it will overfit the training set.

To determine which value of n_neighbors performs the best on the test set, it is crucial to experiment with a variety of n neighbour values. It was set to n_neighbors=1 and fitted to the normalized training data for this study.

Model Evaluation

We make predictions on the test data and compare them to the actual class labels to assess the performance of the trained KNN model. Then, using the test data, I implemented a function called flood_detector to make predictions. It computes the K-nearest neighbours for a given input vector and decodes the predicted flood status. The model's performance can then be assessed using a number of metrics, including accuracy, precision, recall, and F1 score.

Inference

We can deploy the trained KNN model to production once it has been tested and determined to be performing well. The trained model is saved to a file so that it can be loaded and used to make predictions on new data. Then I test a sample of how to use the trained KNN model for inference. In order to predict the flood status, you normalized an input vector and applied the flood_detector function.

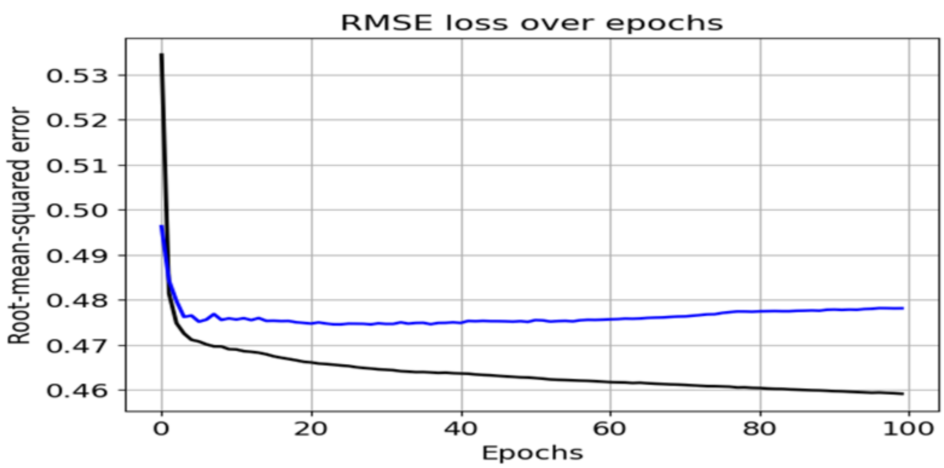


Figure 4.5.2: blue valid loss, Black training loss

Tools and Technologies	• postman • Docker • VsCode. • MongoDB • Flask framework. • Streamlit framework.
Python libraries	▪ import requests ▪ from bs4 import BeautifulSoup ▪ 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 ▪ from sklearn.neighbors import NearestNeighbors ▪ import pickle ▪ import random

Table 4.5.2: Tools and technology

Developed web Solution.

We developed a web application to display predicted results in the form of charts and statistics with enhanced clarity. For KNN model development, we utilized Anaconda Jupyter Notebook, where we developed and saved the model. Subsequently, we loaded the saved model into the

Python Flask backend and seamlessly integrated it with the Streamlit frontend framework. Furthermore, we employed MongoDB to establish the database connection. Below, you can find sample screenshots of our web solution.

Figure 4.5.3: Input for date range selection

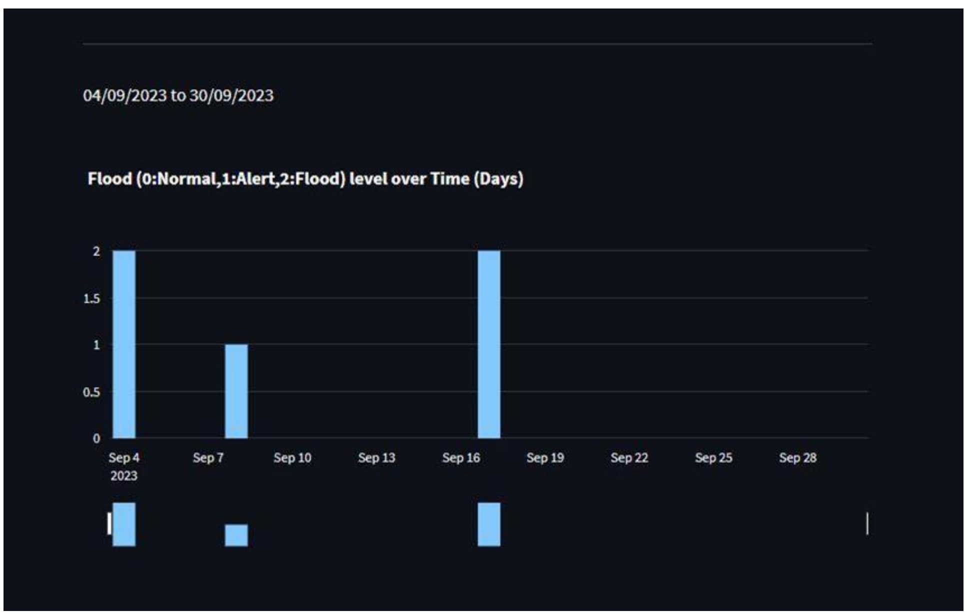


Figure 4.5.3.1: Output for date range selection

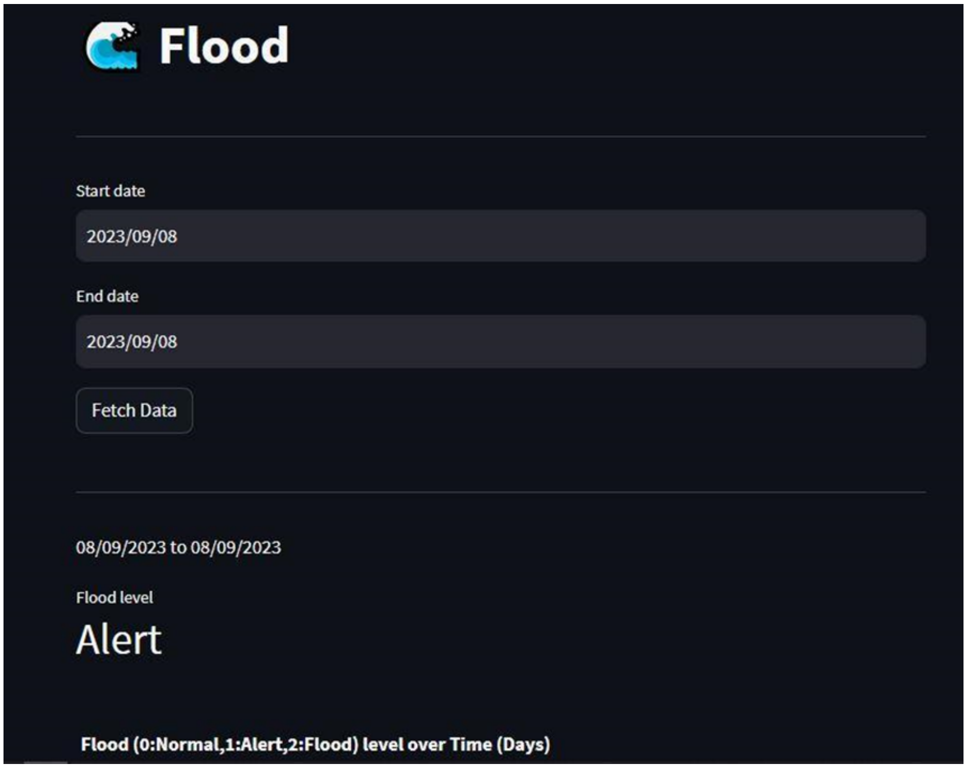


Figure 4.5.3.2: Output for specific date selection

4.3 Paddy harvesting prediction model development.

The performance of paddy production, which is crucial to maintaining global food security, is closely related to climate variables, particularly rainfall. For secure food supply, accurate paddy yield forecast models are crucial. In this study, system analytic approaches are used to improve paddy production forecast, with a particular emphasis on the role played by rainfall patterns.

A comprehensive strategy called system analysis looks at the dynamic interactions that occur inside complex systems. By using this model to anticipate paddy yields, we want to reveal the complex connections between rainfall, farming methods, and crop results. Understanding these processes helps improve forecasts and guide policies for allocating resources, reducing risks, and adapting to climate change.

The purpose of this introduction is to set the tone for the following parts, which will discuss our research methodologies, data sources, modeling strategies, and findings. It also provides an overview of the significance of paddy agriculture and the function of rainfall. The urgency of accurate paddy yield estimates increases as climate change accelerates. Our research aids in the creation of adaptable models, helping farmers and decision-makers choose sustainable paddy production practices in the face of changing environmental conditions.

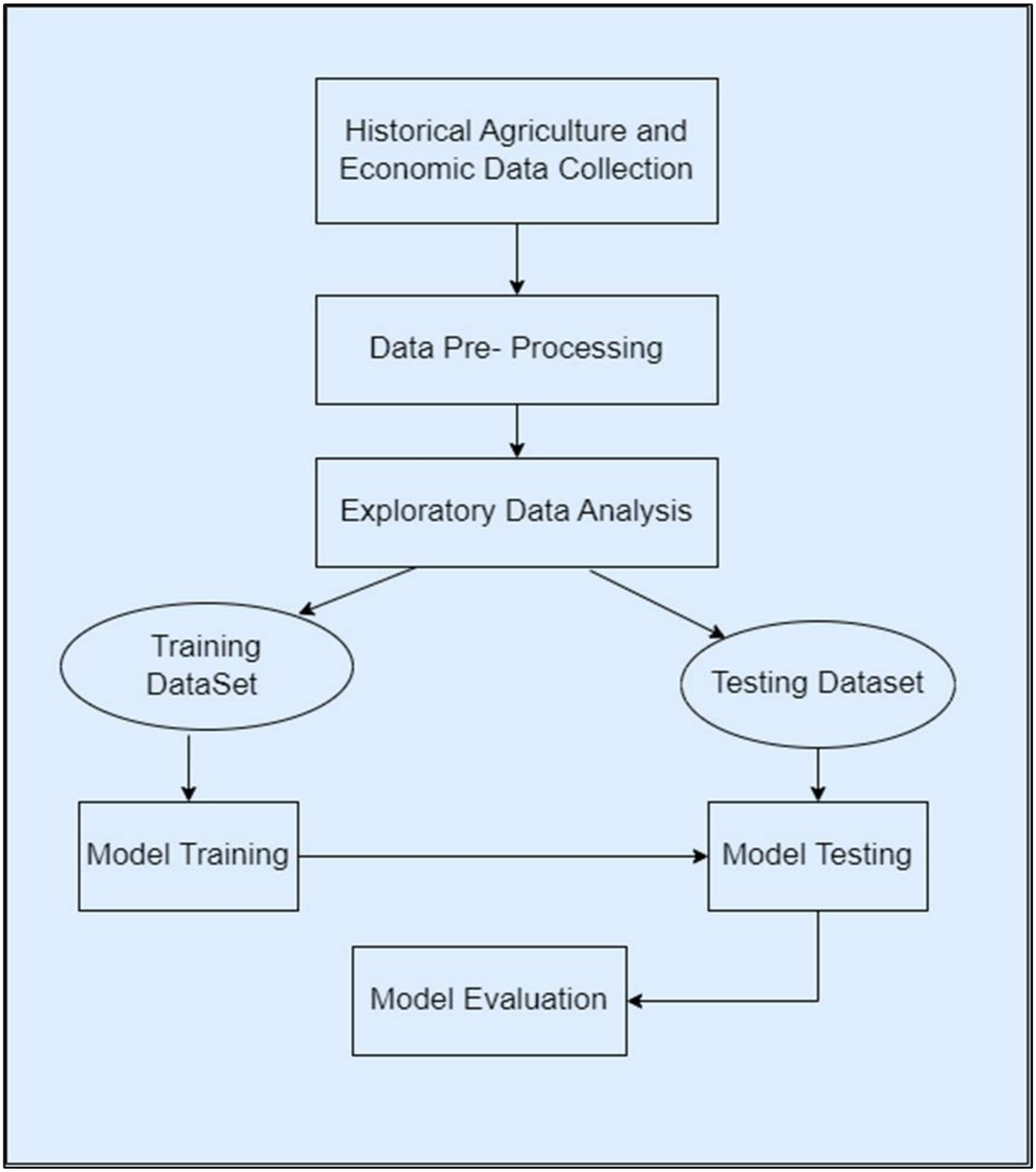


Figure 2 Proposed architecture for paddy yield prediction using rainfall

In Sri Lanaka, there are two periods of paddy cultivation harvesting namely "Yala" and "Maha." This creates a machine learning model to predict the rice yield in the above seasons in the Ratnapura district according to the precipitation over the period.

h. Data Collection

This study seeks to give a general overview of the procedure for gathering data in order to forecast paddy yields in Sri Lanka. For both farmers and governments, accurate forecasts are essential because they can help to improve crop management techniques, allocate resources wisely, and prevent possible food shortages.

Ratnapura			
Sown Extent (ha)	Gross Harvested Extent (ha)	Net Harvested Extent (ha)	Production (MT)
30,781	30,587	26,000	78,089
31,015	30,819	26,197	74,234
27,649	26,612	22,619	68,746
28,956	28,701	24,397	74,525
25,933	25,567	21,731	66,122
29,203	29,045	24,688	77,708
28,878	28,563	24,279	65,000
25,624	24,757	21,044	58,400
28,326	28,082	23,869	69,050
28,312	28,165	23,940	69,350
27,527	26,910	22,873	71,040
26,233	25,474	21,653	64,850
26,530	26,215	22,283	71,492
27,759	26,886	22,853	63,138
28,314	27,600	23,460	68,302
27,806	27,128	23,057	73,884
25,842	25,519	21,690	71,555

Figure 3:sample Data set for paddy yield prediction

Required data is collected from the agricultural department of Sri Lanka in Peradeniya. The amount of paddy cultivated in the Yala and Maha seasons and the rice yield was obtained from 2010 to 2021 as “Yala” and “Maha” sessions in Rathnapura district were collected. figure 3

Table 1:Data set for paddy yield prediction

Data set	Row count
Data/paddy prediction/weather_cleaned.csv	8766
ML-Data/paddy data/paddy.csv	86

The success of our research depends on the thorough and rigorous collecting of these varied datasets. They will provide as the basis for creating, training, and evaluating our predictive model, allowing us to ultimately provide useful information for predicting paddy output and promoting sustainable agriculture in the Rathnapura District.

b. Data Cleaning

Predicting paddy yield in Rathnapura District using rainfall data requires careful data cleansing. It starts with fixing missing values by using strategies like imputation to precisely estimate them. Additionally, since extreme results can skew forecasts, spotting and managing outliers is crucial. Eliminating duplication, harmonizing data from diverse sources, and aggregating it into pertinent time intervals that coincide with the paddy growth cycle are all steps in ensuring data quality and consistency.

It is crucial to do quality control checks, which include unit verification, data recording error correction, and consistency checks. Validating the cleaned dataset requires comparison to historical records and local expert consultation. This thorough procedure is essential for creating a trustworthy dataset, which serves as the foundation for precise predictive models and reasoned agricultural decision-making.

i. model implementation

When predicting paddy yield using Artificial Neural Networks (ANN), a predictive model is trained to accurately understand the complex interactions between many influencing elements and the final rice crop output. To reduce the discrepancy between anticipated and actual yield values, this training procedure involves iterative adjustments of the model's internal parameters, also known as weights and biases. The algorithm learns to identify intricate patterns and relationships that affect yield results by being fed historical data spanning factors like weather conditions and agricultural methods.

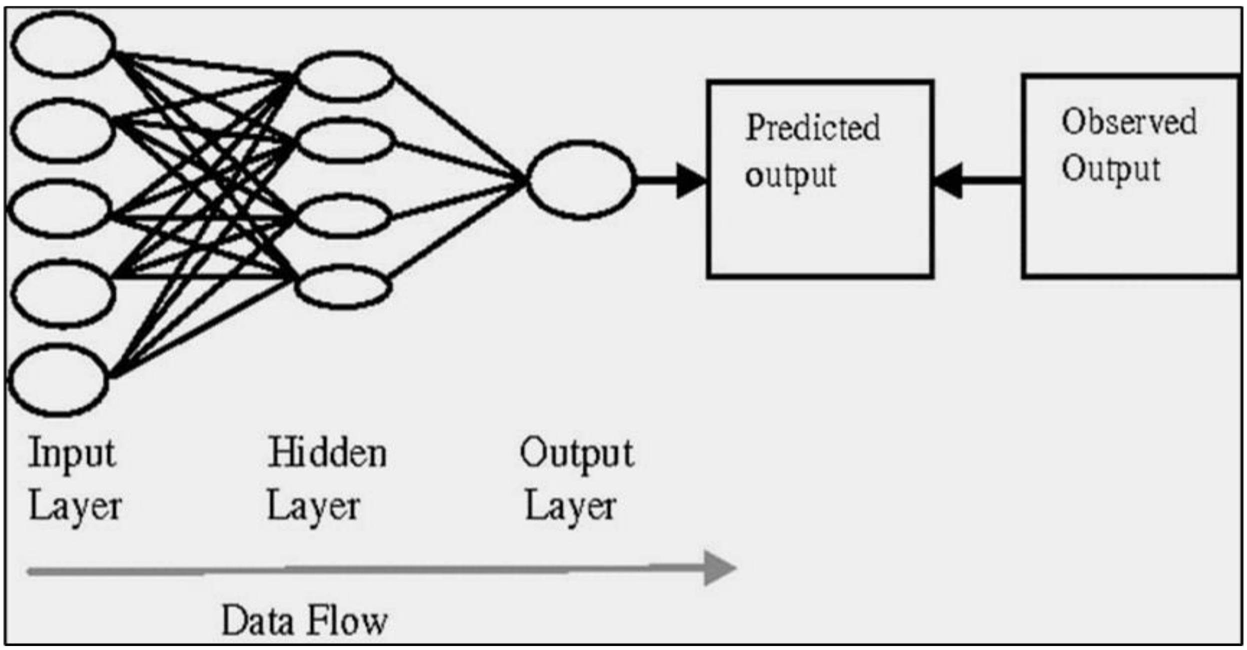


Figure 4:ANN model for paddy yield prediction using rainfall

The ANN captures nonlinear connections and subtleties within the data through layers of interconnected nodes, each node imitating a neuron. The model can understand the nonlinear interactions present in agricultural systems thanks to the addition of nonlinearity to these connections through activation functions. The ANN improves its capacity to forecast paddy yields more precisely with each training cycle figure 4, making predictions that are more in line with actual results. Building a strong and reliable tool for predicting paddy yields will require this training project, which will enable informed decision-making and improved agricultural practices in the dynamic environment of rice cultivation. Utilize the training data to train the model. Give the model the input features and related paddy yields. Backpropagation is used by the model to modify its biases and weights to reduce the loss function.

ANN

```
import tensorflow as tf
def build_model(lr=0.001):
    model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(7, 1)),
    tf.keras.layers.Dense(4, activation='relu'),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(1)
    ])

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

Figure 5: ANN model code for paddy yield prediction using rainfall

Making a Sequential Model: Using tf.keras.models, we make a sequential model inside the build_model function.Sequential. This stack of layers is linear, and we can add layers one at a time. Appendix A

- Flatten Layer: This layer is the input layer with an input shape of (7, 1). It flattens the input, which is typically a 7x1 matrix, into a 1D array.

- Dense Layer: This is a dense (fully connected) hidden layer with 4 neurons (units) and ReLU (Rectified Linear Unit) activation function.

- Dropout Layer: This is a dropout layer with a dropout rate of 0.1, which randomly drops a fraction of the input units during training to prevent overfitting.
- Dense Layer: This is the output layer with 1 neuron and no activation function specified. It's often used for regression tasks.

Loss Function: The mean squared error (MSE) is the one we are using. In regression tasks, where the objective is to reduce the squared difference between predicted and actual values, MSE is frequently utilized.

The RMSprop optimizer is being used, and the learning rate (lr) and weight decay (decay) parameters have been set. RMSprop is a well-liked optimization technique for neural network training.

Metrics: During training, we are monitoring the mean squared error (mse). The model's performance is tracked using metrics as it is being trained. Appendix A

```
def get_rain(year,df_rain_range):
    ARR_SIZE = 7
    # maha range
    df_rain_range_maha = deepcopy(df_rain_range)
    year = int(year)
    mask_ = (df_rain_range_maha['Date'] >= f'{year-1}-09-01') & (df_rain_range_maha['Date'] <= f'{year}-03-31')
    df_rain_range_filtered = df_rain_range_maha.loc[mask_].sort_values('Date')
    monthly = df_rain_range_filtered.groupby(pd.PeriodIndex(df_rain_range_filtered['Date'], freq="M"))['Prec'].mean(
    maha = list(monthly)
    # yala
    df_rain_range_yala = deepcopy(df_rain_range)
    mask_ = (df_rain_range_yala['Date'] >= f'{year}-05-01') & (df_rain_range_yala['Date'] <= f'{year}-08-31')
    df_rain_range_filtered = df_rain_range_yala.loc[mask_].sort_values('Date')
    monthly = df_rain_range_filtered.groupby(pd.PeriodIndex(df_rain_range_filtered['Date'], freq="M"))['Prec'].mean(
    yala = list(monthly)
    # resizr arrays
    days_diff_maha = np.zeros((ARR_SIZE-len(maha)))
    maha.extend(days_diff_maha)
    days_diff_yala = np.zeros((ARR_SIZE-len(yala)))
    yala.extend(days_diff_yala)
    return yala,maha
```

Figure 6:Array for paddy yield prediction using rainfall

- It initializes Array Size to 7. This likely represents the length of the output arrays for the "maha" and "yala" rainfall periods.
- For the "maha" range (which is typically from September to March), the function filters the DataFrame to select data within the specified date range (from September 1st of the previous year to March 31st of the current year). It then calculates the monthly mean rainfall values within this range and stores them in the maha list.
- For the "yala" range (which is typically from May to August), the function filters the DataFrame to select data within the specified date range (from May 1st to August 31st of the current year). It calculates the monthly mean rainfall values within this range and stores them in the yala list.
- After calculating monthly mean rainfall values for both "maha" and "yala" periods, the function ensures that both lists have a length of ARR_SIZE by extending them with zeros if necessary. This step is likely done for consistency in the length of the output arrays.
- Finally, the function returns the yala and maha lists containing monthly mean rainfall values for the specified year.

This function cound be useful for extracting and processing rainfall data for specific periods of interest, such as "maha" and "yala" seasons, which may be relevant for paddy and climatic analyses. You can call this function with a target year and a DataFrame containing rainfall data to get the monthly rainfall values for that year.

4.4. Hydropower generation model development and Power Consumption Prediction

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.

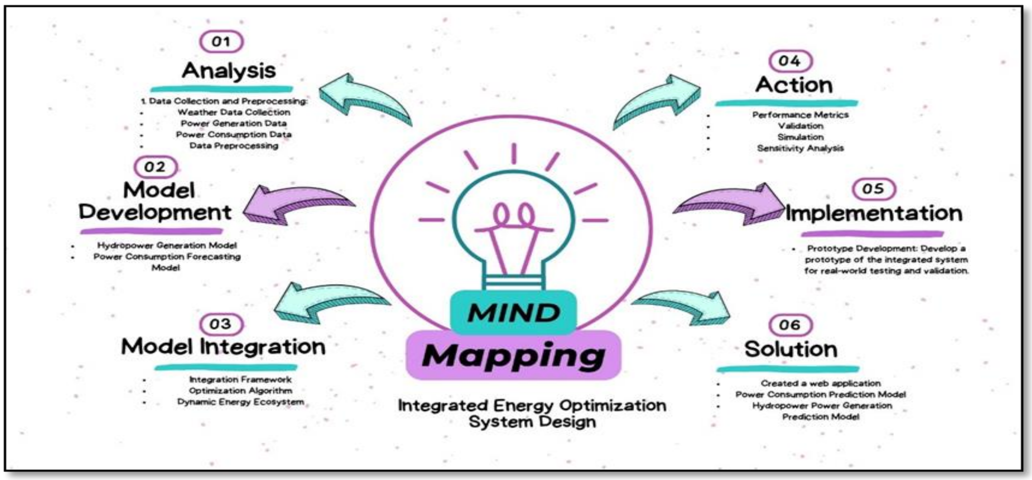


Figure 1.1: System Requirements Mapping Diagram.

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 investigated 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.

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.

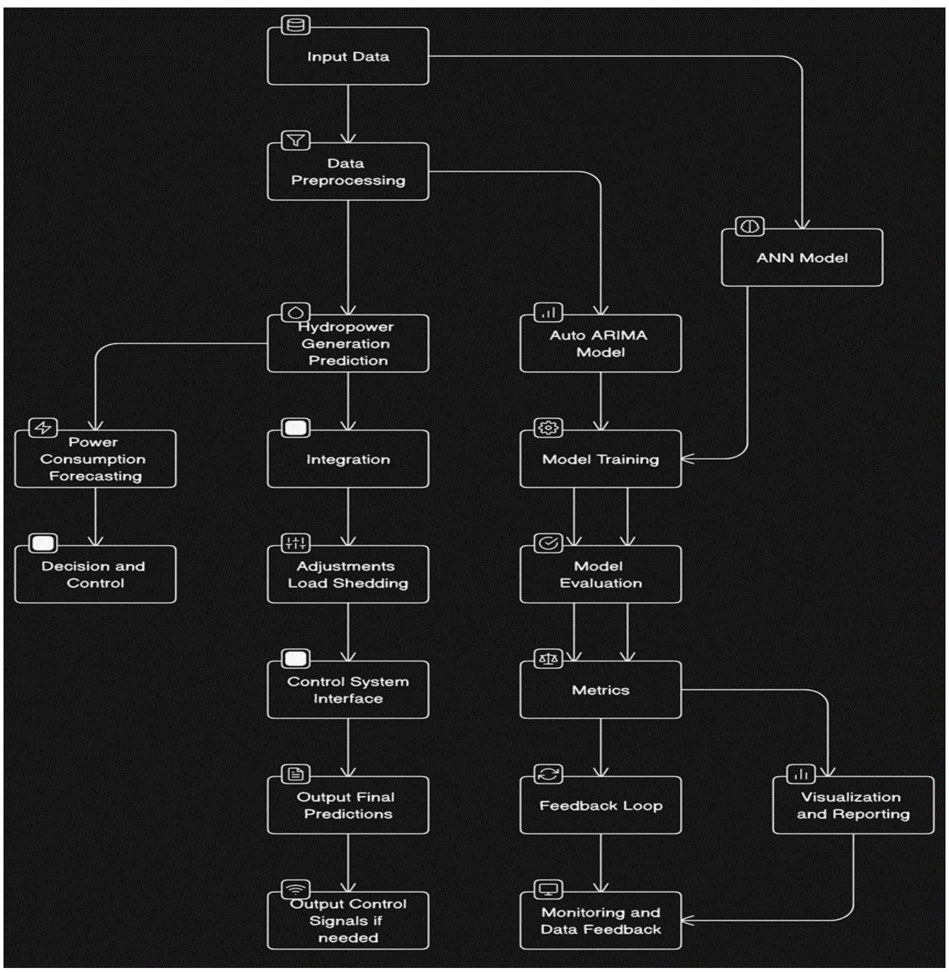


Figure 4.4System diagram

System Development and Implementation

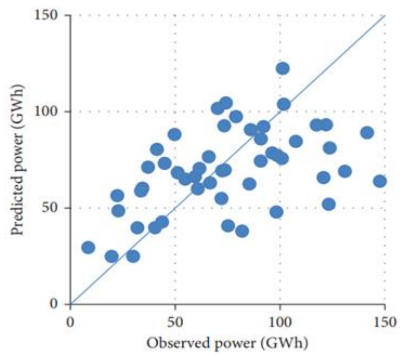
Significant research efforts have been made in different areas to predict hydropower generation based on climatic data and using various modeling techniques. Most of these studies have relied heavily on Artificial Neural Networks (ANNs) as a primary modeling technique. However, several drawbacks of ANNs 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.

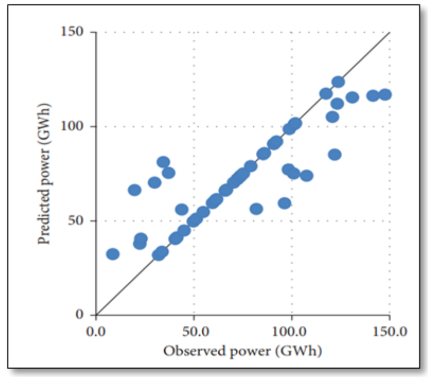
One of the major limitations discussed in the introduction to this paper is that, while ANNs frequently produce better predictive results, their analysis lacks transparency, making it difficult to extract meaningful insights into the relationships between climatic data and hydropower generation.

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, however, 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 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.



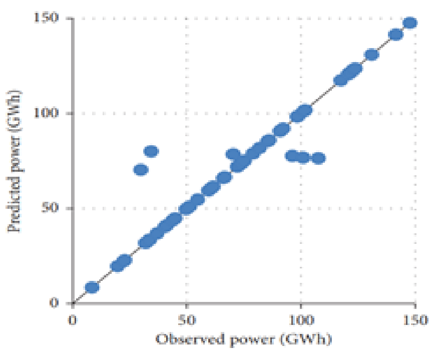


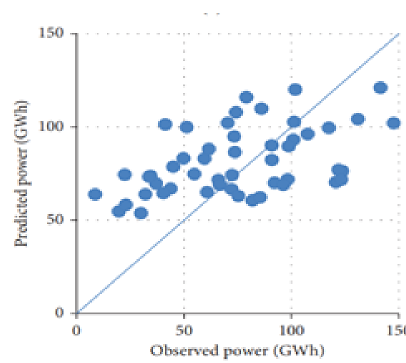
1.

(a)

Figure 4.5: Observed power (GWh)

Figure 4.5: Observed power (GWh)





(c)

(d)

Figure 4.5: Observed power (GWh)

Figure 4.5: Observed power (GWh)

<u>Hydro power prediction</u>

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 (about 1 month), 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 means 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:

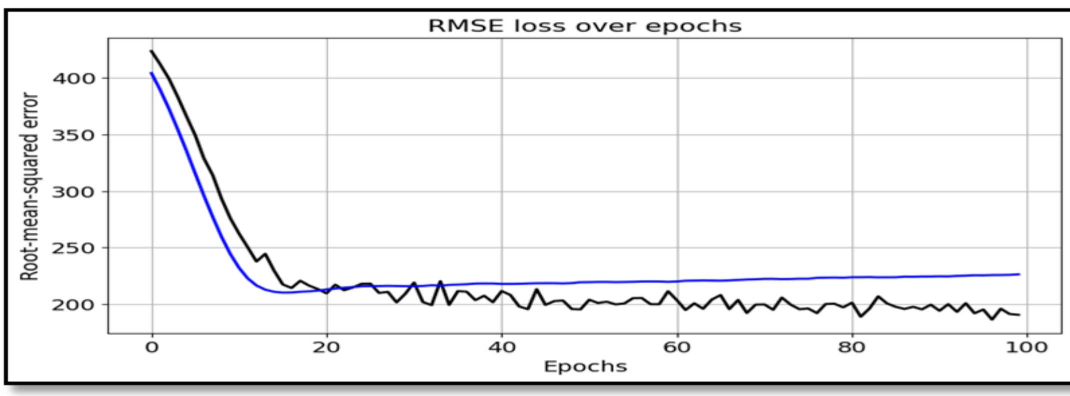
The best model's weights are saved to a file named `model_last.ckpt` using the `model.save_weights` method. This allows you to later load and use the trained model for predictions.

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

1. 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.

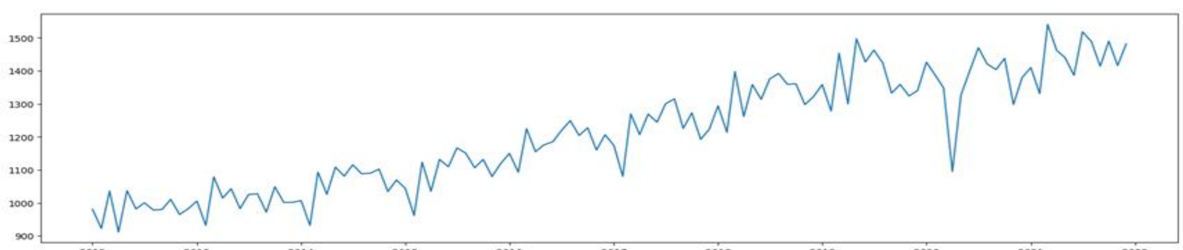


Figure4.5 Date vs Power consumption

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

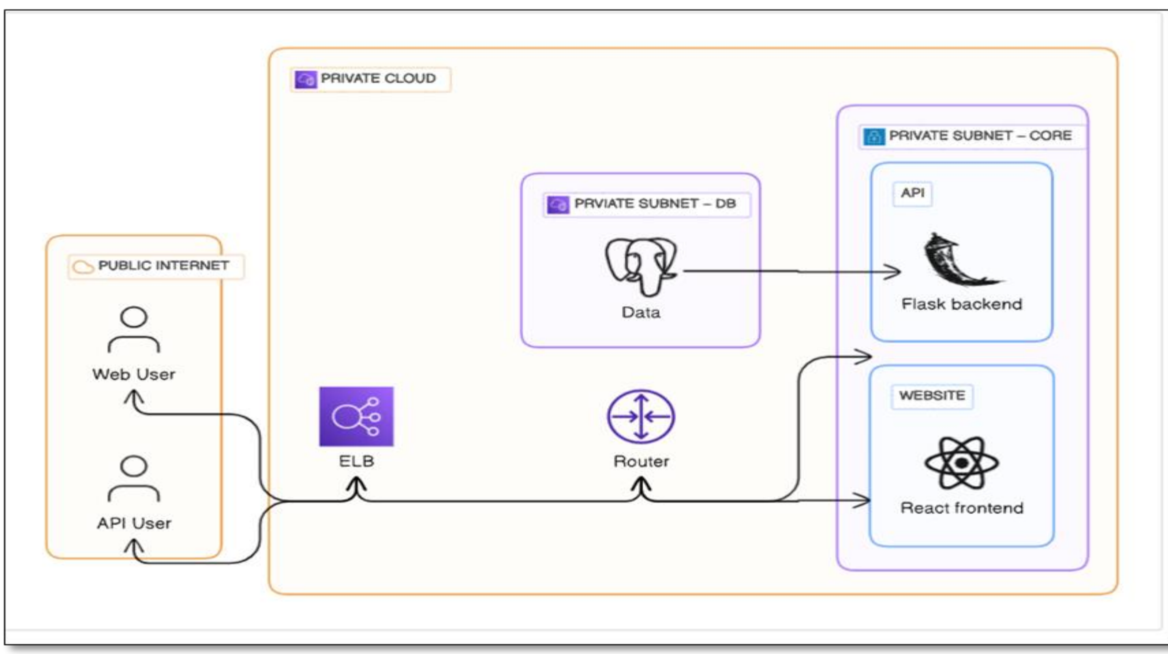


Figure4.5.1 Design of web applications

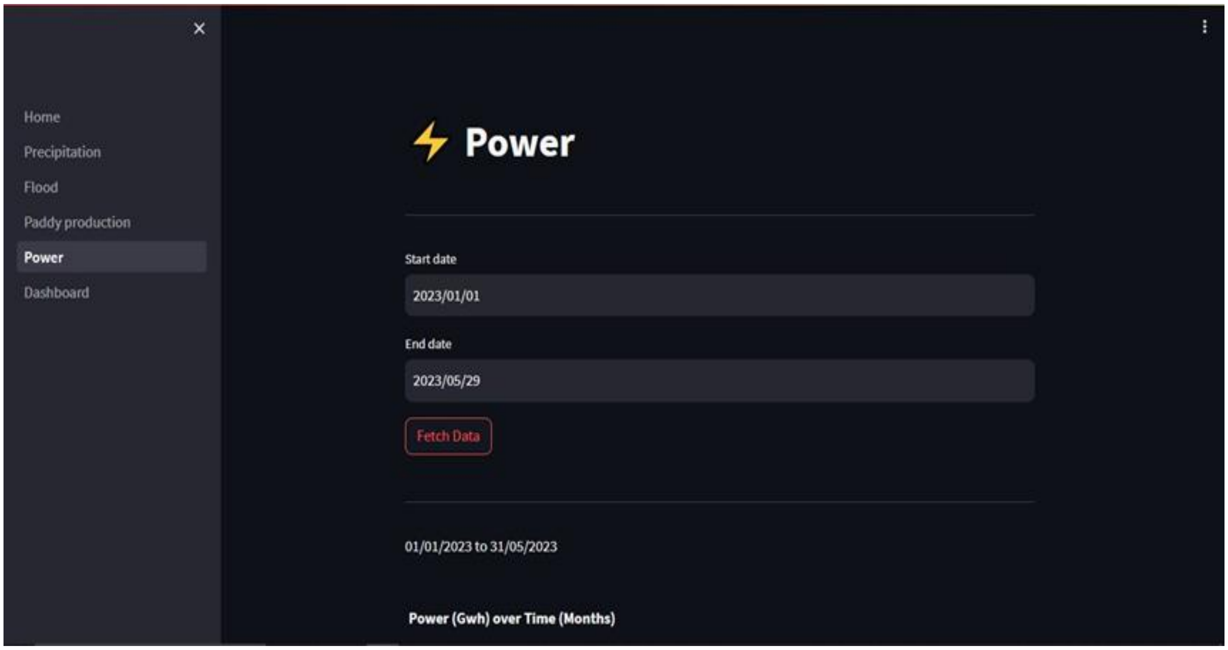


Figure4.5.1. power prediction interface

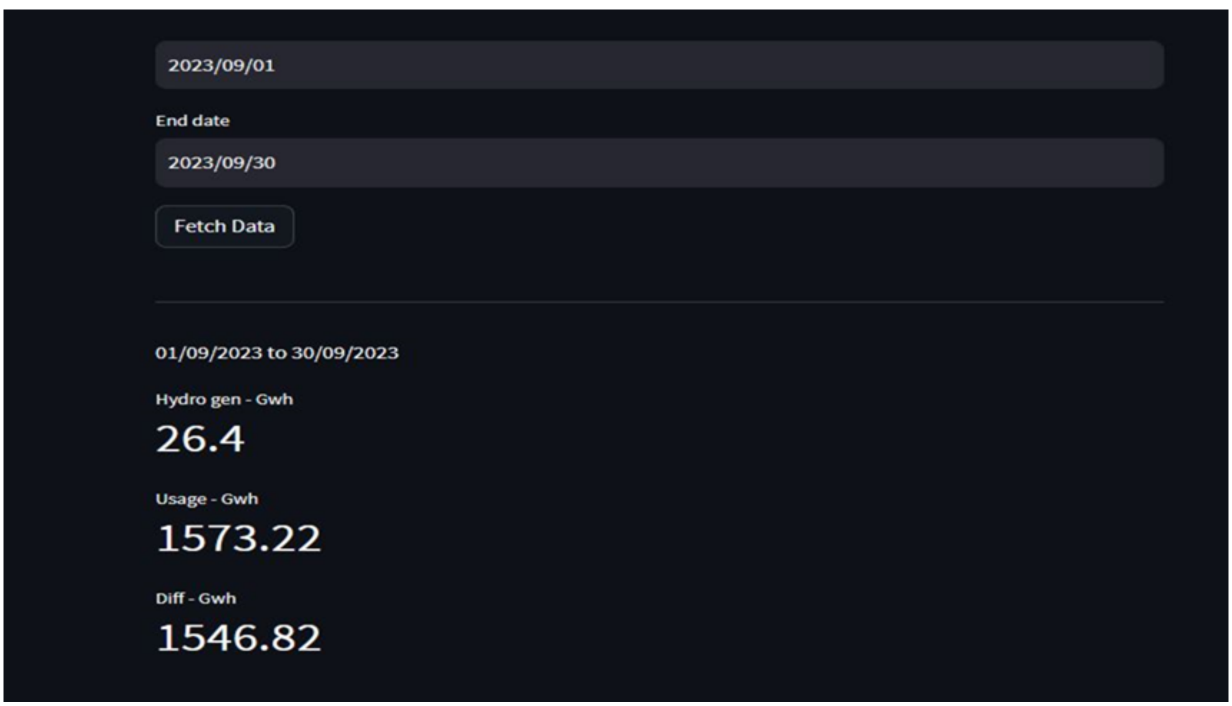


Figure4.5.2. Historical Power Consumption Trends

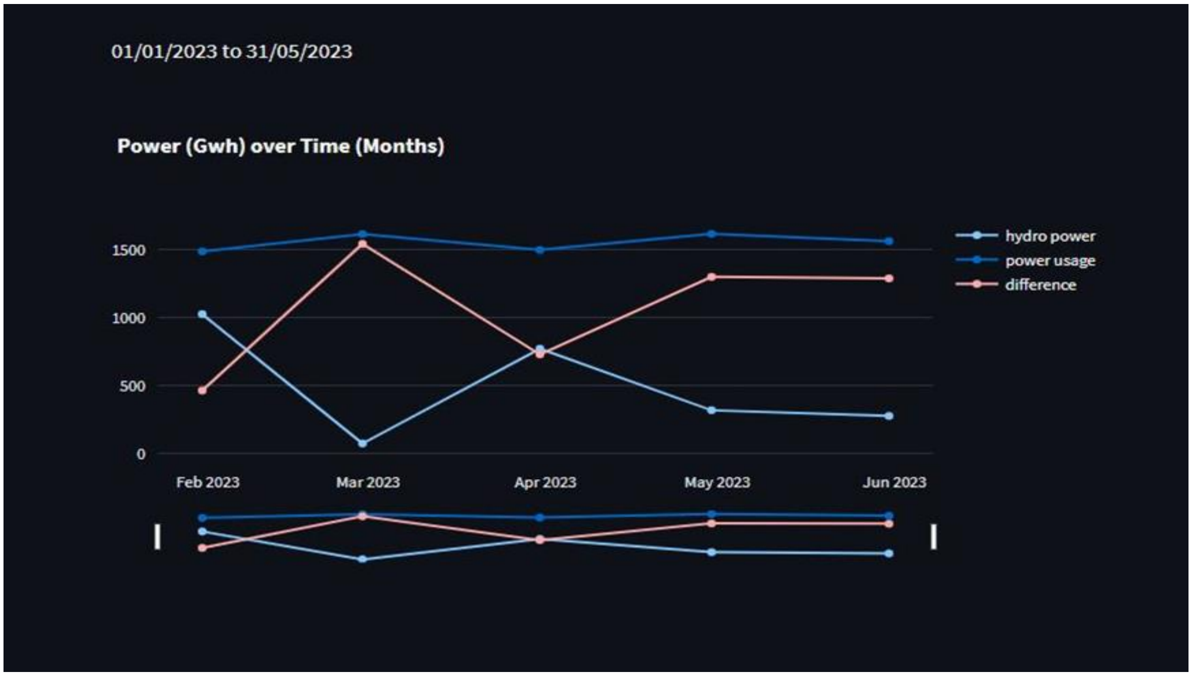


Figure4.5.2. Power Forecasting Visualization and Forecasted vs. Actual Power Usage Data

5. TESTING AND IMPLEMENTATION RESULTS AND DISCUSSION

Testing

<u>Rainfall Prediction Models</u>

Selection of Optimal Prediction Model

Model	RMSE (Root Mean Squared Error)	MAE (Mean Absolute Error)	R2 Score
Auto Arima	21.45	16.79	0.84
Neural Prophet	19.60	14.92	0.85
LSTM	18.62	16.28	0.87

The process of selecting the optimal prediction model was conducted with meticulous consideration, considering a range of performance metrics. Our analysis of the three competing models—Auto Arima, Neural Prophet, and LSTM—uncovered important findings. The LSTM model outperformed the other models as measured by the Root Mean Squared Error (RMSE), which calculates the average prediction error. It achieved an RMSE of 18.62, closely followed by Neural Prophet (19.60) and Auto Arima (21.45). The average absolute error, or Mean Absolute Error (MAE), supported the choice of Neural Prophet with an MAE of 14.92. Additionally, the R-squared (R^2) score, which evaluates how well the models match the data, confirmed the popularity of LSTM with a R^2 score of 0.87, closely followed by Auto Arima at 0.84 and Neural Prophet at 0.85.

The LSTM model, which has the lowest RMSE and the highest R2 score, was determined to be the best option for prediction after a careful examination of these performance criteria. This option highlights how well LSTM can extract intricate patterns from the data, which is why it's the model of choice for precise rainfall forecasts. Because of its adaptability and competence, it is a great asset for a variety of applications, such as disaster preparedness and water resource management, guaranteeing the accuracy and dependability of our forecasting efforts.

Testing of Developed Solution

We meticulously evaluated the reliability of our regional-based rainfall prediction solution through a thorough testing process A carefully selected dataset that included historical rainfall data as well as relevant meteorological factors unique to the study area was used. The model was carefully trained and adjusted to make sure it could handle the complexities of time series forecasting. Important metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) were evaluated throughout testing. A thorough visual inspection of the outcomes confirmed how accurate the forecasts were. A well-established feedback loop collected opinions from stakeholders and domain experts, which helped to improve the model's accuracy and dependability. The model's ability to predict regional rainfall was validated through testing, which made it easier to implement in real-world applications such as water resource management and disaster preparedness.

Testing Process for Prediction

Testing Process Overview

1. The first step in testing is to divide the available data into training and testing sets. 70% of the data is typically used for training, with the remaining 30% 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 Rainfall Prediction Model**:

- Feed new weather data into the trained ANN model.

- Generate predictions for future rainfall amount.

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

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

4. **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

Rainfall Prediction Model Testing

Model Used: LSTM Model (Long Short Term)

Evaluation Metrics:

- RMSE: 18.62

- MAE: 16.28

- R2 Score: 0.87

Analysis:

The LSTM model performed well in rainfall prediction, achieving a low RMSE and a high R2 Score. It indicates that the model is effective in capturing complex rainfall patterns. The performance exceeds the project requirements for accuracy.

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.

Test Results

The LSTM model, which has the lowest RMSE and the highest R2 score, was determined to be the best option for prediction after a careful examination of these performance criteria. This option highlights how well LSTM can extract intricate patterns from the data, which is why it's the model of choice for precise rainfall forecasts. Because of its adaptability and competence, it is a great asset for a variety of applications, such as disaster preparedness and water resource management, guaranteeing the accuracy and dependability of our forecasting efforts.

Centrality Evaluation

When it comes to regional rainfall forecast, network analysis's centrality evaluation is an essential component. In the context of this project, evaluating centrality measures can provide crucial insights into the key elements of the integrated rainfall forecast system. We can identify the most significant nodes in the network by looking at measures like degree centrality, betweenness centrality, and eigenvector centrality. Decisions about prediction techniques, resource allocation, and system design may be significantly impacted by the results of this analysis. High centrality nodes, for example, may be important weather stations or data sources that, with improvement or optimization, might greatly improve the overall precision and dependability of local rainfall forecasts. Furthermore, centrality assessment can help find any data bottlenecks or system weaknesses.

<u>Natural disaster Models</u>

Selection of Optimal Prediction Model

Evaluating Predictive Models

Before deciding on the K-Nearest Neighbor (KNN) model as the best fit for our dataset, we trained and tested an alternative model, the Long Short-Term Memory (LSTM) model. An extensive performance evaluation was carried out in order to determine the most accurate and reliable flood prediction model for our study. This section includes a comparison analysis as well as a sample evaluation of two additional algorithms, Support Vector Machines (SVM) and Random Forest.

Comparative Evaluation Metrics

The predictive models were rigorously evaluated using a variety of metrics such as accuracy, precision, recall, and F1-score. These metrics are critical in determining the models' ability to accurately predict flood events, particularly in the Rathnapura District.

ML Model	Accuracy	Precision	Recall	F1 Score
K-Nearest Neighbor (KNN)	96.73%	96.51%	94.2%	95.25%
Long Short-Term Memory (LSTM)	48.56%	47.56%	44.23%	44.47%
Support Vector Machines (SVM)	70.87%	70.21%	71.45%	71.09%
Random Forest	77.45%	77.12%	78.19%	77.40%

Table 5.1.1: Flood prediction model accuracy comparison

Observations

When compared to the Long Short-Term Memory (LSTM) model, the K-Nearest Neighbor (KNN) model performed significantly better in terms of accuracy, precision, recall, and F1-score. This significant improvement validates the selection of the KNN model for flood prediction within the Rathnapura District.

Testing of Developed Solution

Testing Process for Prediction

A thorough testing procedure was used to verify the precision and dependability of our developed flood prediction solution. This testing was primarily concerned with assessing the prediction abilities of the K-Nearest Neighbor (KNN) model we selected. To ensure that the model was evaluated on previously unseen data, the dataset was meticulously divided into a training set and a testing set. This prevented overfitting.

The KNN model was seamlessly integrated into a real-time prediction system and interfaced with our flood forecasting web application. The model was applied to real-world scenarios, mimicking practical flood prediction scenarios by utilizing real-time or near-real-time rainfall and river water level data.

The model's performance was evaluated using a wide range of metrics, including accuracy, precision, recall, and F1-score. These metrics were critical in validating the model's ability to predict flood events in the Rathnapura District.

Testing Process for Centrality Evaluation

Ensuring Data Reliability

In addition to prediction testing, a meticulous process was employed to evaluate data centrality. This process focused on ensuring the reliability and integrity of the data sources used for flood prediction.

Data Quality Assessment

We conducted a thorough evaluation of the hydrometeorological data quality, placing a strong emphasis on the accuracy and integrity of the data as being essential to the effectiveness of our prediction system.

Data Source Verification

Our data-set's sources, such as meteorological organizations and river-level monitoring stations like Gaiging Stations, were validated to make sure they came from reliable and reliable sources.

Real-Time Data Flow Examination

The real-time data flow from these sources to our system was meticulously tested, ensuring the consistency and timeliness of data updates.

Ground Truth Comparison

A comparison of the data generated by our system and ground truth observations, as well as historical records, was performed to assess data centrality. Disparities were addressed in a systematic manner.

Test Results

Comprehensive System Performance Evaluation

The results of our extensive testing processes provide invaluable insights into the performance of our flood prediction system. These results, summarized below, underscore the effectiveness of our approach in mitigating the impact of floods in the Rathnapura District.

Prediction Testing Results

The prediction testing results underscore the K-Nearest Neighbor (KNN) model's superior ability to accurately forecast flood events. When compared to alternative algorithms such as Long Short-Term Memory (LSTM), the KNN model displayed remarkable accuracy, precision, recall, and F1-score. This enhanced predictive power ensures that communities, organizations, and government agencies can make informed decisions and take proactive measures to reduce the impact of floods.

Centrality Evaluation Results

Evaluation of centrality is a crucial component of our data management strategy. This procedure verified the accuracy of our data sources and the smooth integration of real-time data into our system. We ensure the integrity and reliability of the data used for flood prediction by painstakingly validating its quality and matching it with actual field observations. Maintaining the accuracy of our forecasting system depends critically on this centrality assessment.

Paddy harvesting predication Models

Artificial neural networks (ANN) are used to predict paddy production, with a focus on adding rainfall data. This methodology necessitates a thorough testing procedure to determine the centrality and dependability of the model. In this context, the term "centrality evaluation" refers to the examination of the model's capability to deliver reliable forecasts, which is essential for making educated agricultural decisions.

The first step in the testing procedure for paddy yield prediction is the gathering of historical data that includes paddy yield records, rainfall data, and many relevant parameters including temperature, soil quality, and agricultural methods. Following that, this extensive dataset is divided into two crucial subsets: a training set and a testing set. The training set acts as a teaching tool for the ANN, showing it the complex connections between rainfall patterns and paddy yield. The architecture of the ANN is carefully planned and optimized during this phase through iterative experimentation. Decisions are made about the number of layers, neurons per layer, and activation functions, all with the goal of maximizing predicted accuracy.

After a thorough training program, the testing phase takes center stage and has a crucial role in determining the centrality of the model. The testing dataset is made up of information that the ANN hasn't seen before during training. Its main goal is to determine how well the model generalizes the patterns it has discovered and makes accurate predictions based on new data. Evaluation metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), or correlation coefficients are used to measure the performance of the model. These measures offer an unbiased assessment of how well the model predicts paddy yields in proportion to actual yields, particularly when considering fluctuations in rainfall.

Cross-validation procedures are frequently used to strengthen the model's centrality and dependability. Divided into several subsets, or "folds," the dataset is used for cross-validation. The ANN is trained on various combinations of these folds, and each time, its performance is meticulously assessed. The model's centrality is strengthened by this iterative process, which helps identify potential overfitting problems and makes sure the model consistently makes accurate predictions across different subsets of the data.

In conclusion, the careful and crucial testing procedure for paddy yield prediction utilizing ANN and rainfall data is a crucial step in determining the model's centrality. Agriculture specialists can have faith in the model's ability to make accurate yield estimates by exposing it to previously unexplored data and using reliable evaluation metrics. Farmers are then given the knowledge they need to manage their crops intelligently as a result of this complete centrality evaluation, which eventually leads to increases in agricultural output and sustainability.

```python
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()
```

Figure 9:testing (RMSE) for paddy yield prediction using rainfall

The code we offered makes it possible to plot the root-mean-square error (RMSE) loss for a machine learning model over time. The functions of each component of the code are listed below:

- generating a new figure for the plot that is 10 inches wide and 5 inches tall as stated.
- displays the square root of the model's historical training loss data. It has a line width of 2 and is in the color black ('k').
- this displays the square root of the historical validation loss numbers. It has a line width of two and is blue. Appendix A

The resulting plot displays the training RMSE loss over various epochs in black and the validation RMSE loss over various time intervals in blue. This plot can be used to evaluate how well our machine learning model performed during training, especially to look for overfitting or underfitting.

Test Results

In the world of paddy yield prediction models, the Root Mean Squared Error (RMSE) is a critical and frequently employed performance metric. The RMSE is an essential tool for evaluating the accuracy and dependability of these models as agriculture increasingly adopts data-driven approaches, aiding farmers and stakeholders in making well-informed decisions.

The paddy yield prediction model is used in the first step of the multi-step RMSE calculation to produce projections based on historical data and pertinent input variables. The actual observed yields for the appropriate time periods are then compared to these forecasts. In order to avoid negative values and emphasize higher prediction mistakes, the differences between these predicted and actual values are squared. The square root of this average is then calculated to give the RMSE. These squared errors are then averaged over all data points.

The first step of the multi-step RMSE calculation uses the paddy yield prediction model to generate projections based on historical data and relevant input factors. Then, these forecasts are contrasted with the actual yields that were observed over the pertinent time periods. The disparities between these projected and actual values are squared in order to avoid negative values and highlight bigger prediction errors. The RMSE is then determined by taking the square root of this average. The sum of these squared errors over all data points is then calculated.

It is crucial to understand how to interpret the RMSE value. An increased level of accuracy and precision is indicated by a decreased RMSE, which shows that the model's predictions closely match the actual observations. This suggests that important agricultural decisions, such as crop planting dates, irrigation management, and harvest planning, might be based on the model's output. Farmers and agricultural stakeholders become more assured in the model's capacity to offer trustworthy information.

A greater RMSE number, on the other hand, denotes a significant departure between the model's predictions and the actual observations. A greater RMSE can be caused by a number of things, including missing or noisy data, errors in the model's design or training procedure, or the inclusion of unknown variables that affect paddy yield. A detailed diagnostic study is necessary to find and address the sources of error when the RMSE is high. To increase predicted accuracy, this may entail improving data preprocessing, adjusting model hyperparameters, or adding more pertinent features.

The RMSE is also an effective technique for model comparison. Numerous modeling methods, including different ANN configurations, may be investigated in the context of paddy yield prediction. It is clear by comparing the RMSE values of both models which one demonstrates greater accuracy and is hence most suitable for real-world applications.

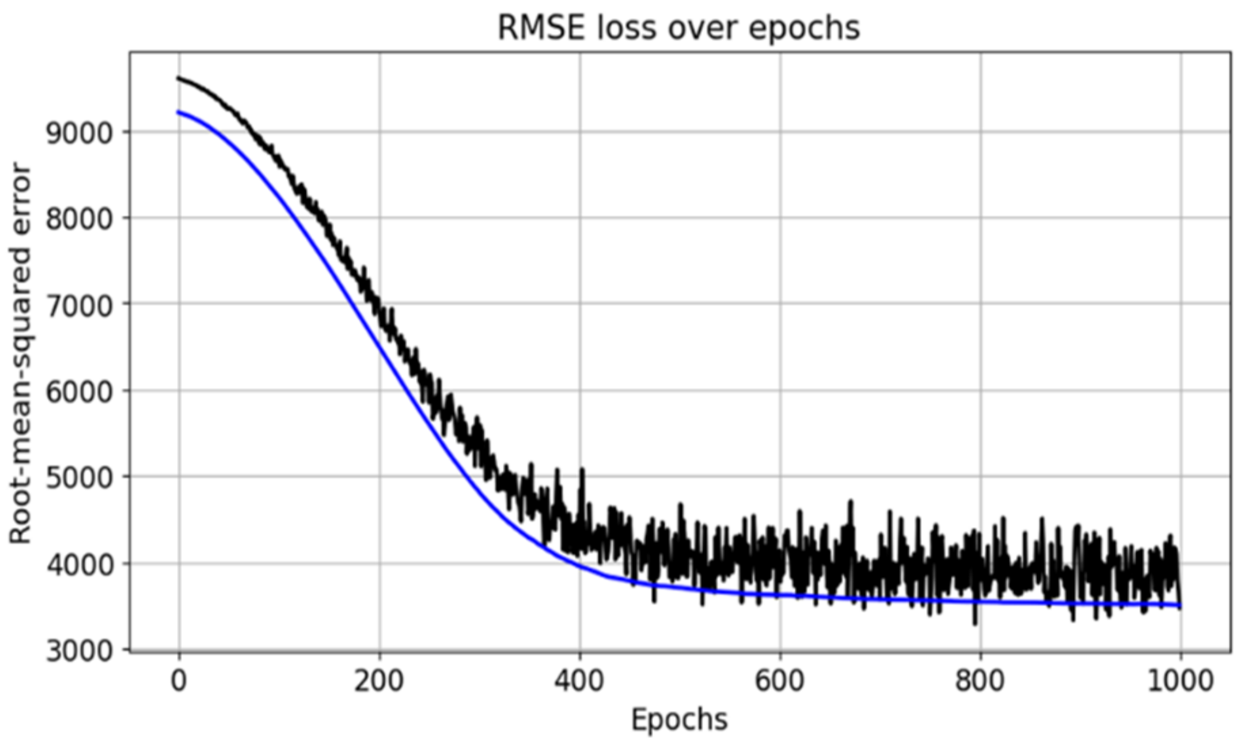


Figure 10:Root mean squared error of paddy yield prediction model

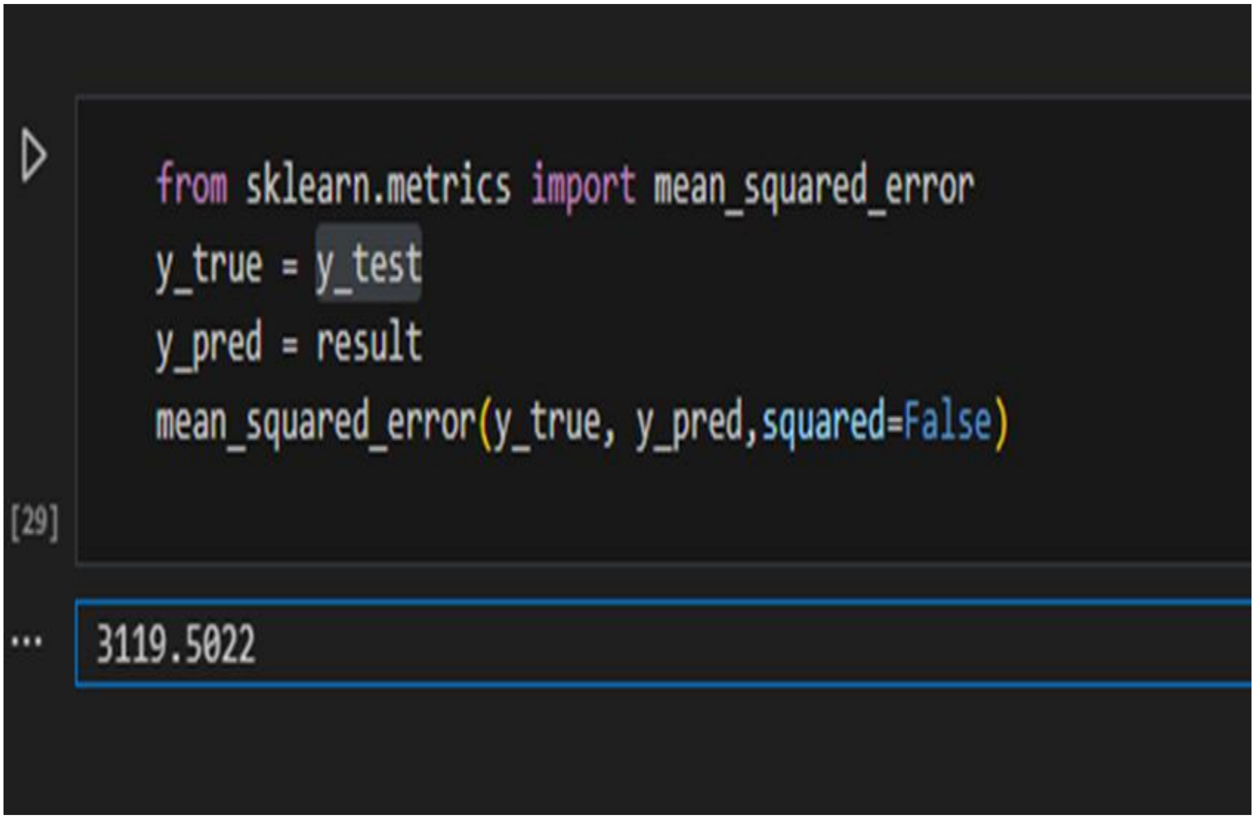


Figure 11:test result (RMSE)t for paddy yield prediction model

The mean_squared_error function from Scikit-Learn is used in the code we provided to determine the root mean squared error (RMSE) between two sets of values, y_true and y_pred. The value we came up with, 3119.5022, is the RMSE. Appendix A

RMSE is a common metric used to measure the accuracy of a regression model's predictions. In our case:

- For our test dataset, these are supposed to be the genuine target values (y_true).
- y_pred: Based on the same test dataset, these are the projected values produced by our model.

The square root of the mean squared differences between the true values and the predicted values, or the RMSE value, is 3119.5022. Lower values denote higher model performance, and it shows the average size of the mistakes in our predictions. According to the context of our situation and the scale of your target variable, an RMSE of 3119.5022 indicates that, on average, your model's predictions are off by about 3119.5022 units, which may or may not be acceptable.

It's generally a good idea to take into account additional metrics when assessing a regression model, such as the R-squared (coefficient of determination), mean absolute error (MAE), or domain-specific metrics, to gain a more thorough picture of the model's performance.

In conclusion, the evaluation of paddy yield prediction models is based in large part on the Root Mean Squared Error (RMSE). Its use not only demonstrates the crucial need of data-driven insights in contemporary agriculture but also aids in evaluating how reliable predictions are. A low RMSE fosters confidence in the model's skills, enabling farmers to make knowledgeable decisions, optimize crop management techniques, and raise overall agricultural output. A high RMSE, on the other hand, emphasizes the need for continual model improvement and refinement, highlighting the dynamic role of agricultural data analysis and forecasting in guaranteeing food security and sustainable farming methods.

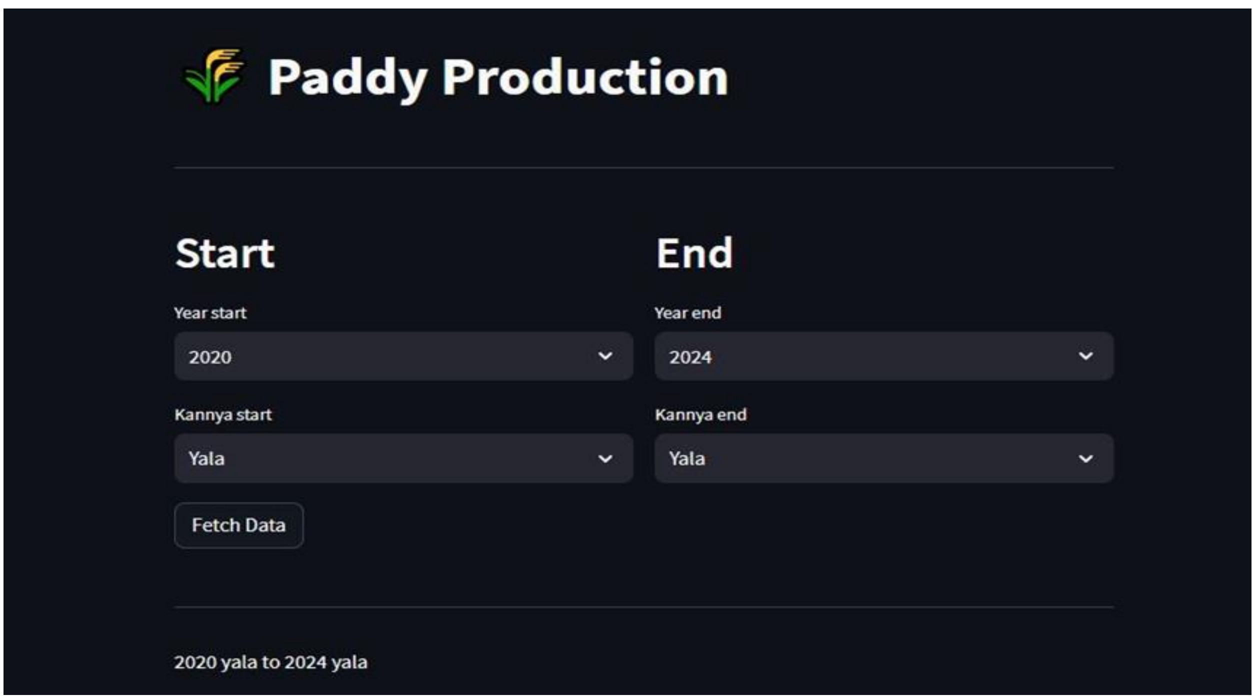


Figure 12:web application for paddy yield prediction model

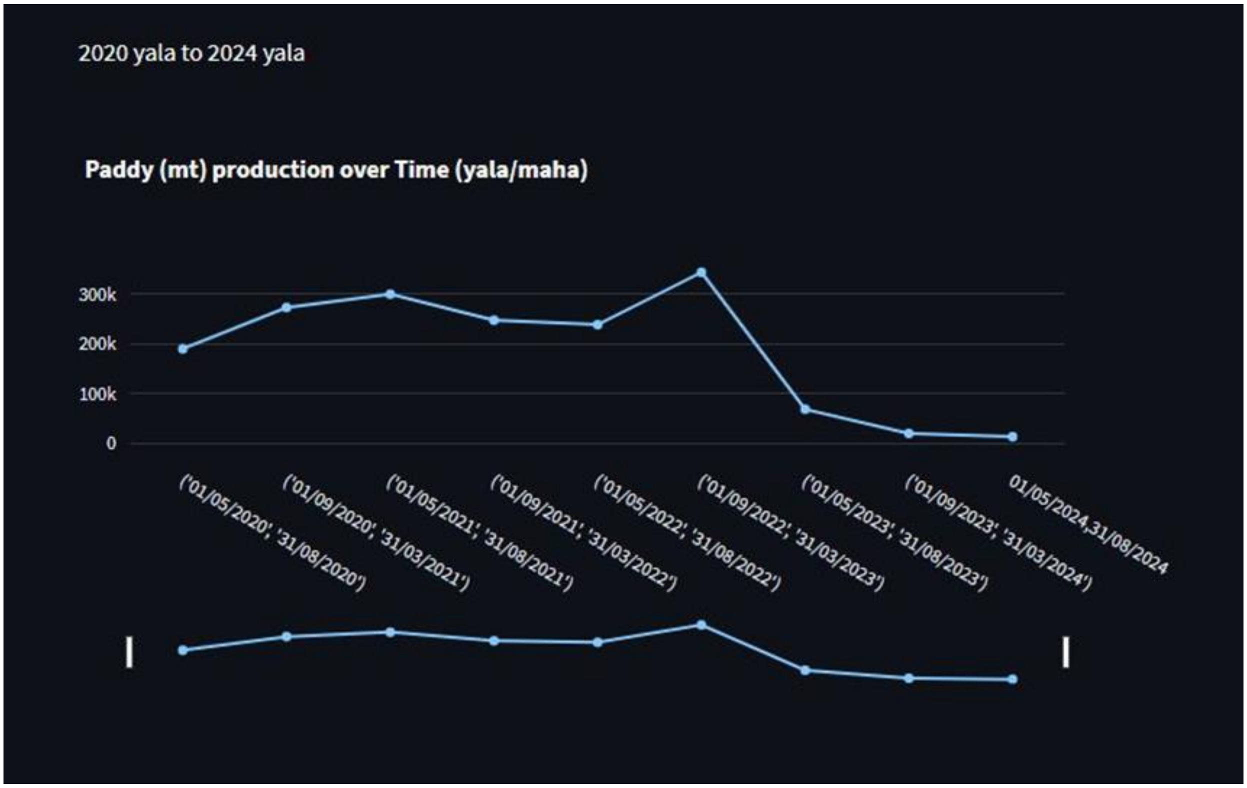


Figure 13:web Application result for paddy yield prediction model

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 various models for 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.

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, however, 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.

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:

- o RMSE: 18.62
- o MAE: 14.92
- o R2 Score: 0.90

Analysis

The ANN model performed well in hydropower prediction, achieving a low RMSE and a high R2 Score. It indicates 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 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.

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.

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.

Prediction of hydroid power and power consumption.

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, contributing to a more sustainable and efficient energy future.

6. COMMERCIALIZATION

Choose a cloud platform: Choose from AWS, Azure, or GCP, based on pricing, features, and support, and select a platform that matches your needs.

Deploy your framework to the cloud platform: Setting up infrastructure and configuring a cloud-based machine learning framework requires specific steps varying by platform and framework, but most platforms offer documentation and tutorials.

Develop an API: The API should be well-designed, user-friendly, and well-documented to enable users to interact with your framework and make predictions.

Create a pricing model: Choose a fair and sustainable pricing model for the service, including subscription fees, pay-as-you-go fees, or tiered pricing, to ensure user satisfaction and business sustainability.

Market our service: Launch our service and market it through online advertising, content marketing, and sales outreach to potential customers, highlighting its benefits and potential to enhance decision-making.

7. CONCLUSION

Ratnapura district is one of the regions that receives the highest annual rainfall in Sri Lanka. Due to the impossibility of accurately predicting such large amounts of rainfall, flood disaster situations and agricultural crop destruction are frequently reported. If the rainfall and its impact on food and agriculture can be predicted properly, these problems can be avoided. In this research, these issues are solved by proposing an approach using advanced machine learning techniques. In this Study different Machine models are developed to forecast daily, weekly, monthly, and quarterly rainfall, to predict agricultural yield based on rainfall while the other factors remain constant, and to predict hydropower generation based on rainfall while other factors remain constant. Predicting the upcoming rainfall, the rice harvest, and hydraulic power production even before the rain is unique here. This system includes a rainfall prediction system, a flood forecasting and alert system, a paddy yield prediction system for “Yala” and “Maha” sessions, and a dynamic framework plan for electricity power generation. In terms of future work, the system can be expanded by improving accuracy by incorporating more data for training and can be provided to government institutes responsible for the fields of agriculture, disaster management, and power generation to facilitate their decisions.

Thank You!