KOCHI UNIVERSITY OF TECHNOLOGY

Implementation and its applications of novel Cascaded Adaptive Network-based Fuzzy Inference Systems

by

Namal Rathnayake

Advisor
Professor Yukinobu Hoshino

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Engineering Faculty
School of System Engineering

March 2023

Abstract

The adaptive neuro-fuzzy inference system (ANFIS) is employed in a vast range of applications because of its smoothness (by Fuzzy Control (FC)) and adaptability (by Neural Network (NN)). Although ANFIS is better in non-linear optimization, two major loopholes must be addressed thoroughly. They are the curse of dimensionality and computational complexity. To overcome these complications, a novel usage of the ANFIS model is proposed as Cascaded-ANFIS. As the primary source of this algorithm, a general two-input one-output ANFIS algorithm is used. The novel algorithm has two main modules called pair selection and training model. Pair selection selects the best match for the inputs, while the training module generates the output. The cascaded behaviour of the novel algorithm causes additional iterations to advance to the best solution. Even though the number of parameters that need to be adjusted increases at each additional iteration, the algorithm's complexity may stay stable.

The reliability and excellent performance of the Cascaded-ANFIS were tested by performing various experiments. Initially, two-hybrid state-of-the-art algorithms are used to compare the performance of the novel algorithm, namely, Particle Swarm Optimization based ANFIS (ANFIS-PSO) and Genetic Algorithm based ANFIS (ANFIS-GA). Furthermore, individual performance is presented for seven publicly recognized data sets. The results have demonstrated that, for some data sets, the Root means square error (RMSE) can be a minimum of 0.0001. Furthermore, the novel algorithm was used in several practical applications.

Automated fruit identification is always challenging due to its complex nature. Usually, the fruit types and sub-types are location-dependent; thus, manual fruit categorization is still a challenging problem. As the first application in real-time, the Cascaded-ANFIS was used on image classification tasks. Literature showcases several recent studies incorporating the Convolutional Neural Network-based algorithms (VGG16, Inception V3, MobileNet, and ResNet18) to classify the Fruit-360 dataset. However, none is comprehensive and has not been utilized for the total 131 fruit classes. In addition, the computational efficiency was not the best in these models.

A robust, comprehensive, novel study is done to identify and predict the entire Fruit-360 dataset, including 131 fruit classes with 90,483 sample images. An algorithm based on the Cascaded Adaptive Network-based Fuzzy Inference System (Cascaded-ANFIS) was effectively utilized to achieve the research gap. Colour Structure, Region Shape, Edge Histogram, Column Layout, Gray-Level Co-Occurrence Matrix, Scale-Invariant Feature Transform, Speeded Up Robust Features, Histogram of Oriented Gradients, and Oriented FAST and rotated BRIEF features are used in this study as the features

descriptors in identifying fruit images. In this study, the algorithm was validated using two methods: iterations and confusion matrix.
The results show that the proposed method has a relative accuracy of 98.36%. The Fruit-360 dataset is unbalanced; therefore, the weighted precision, recall, and F1-Score were calculated as 0.9843, 0.9841, and 0.9840, respectively. In addition, the developed system was tested and compared against the literature-found state-of-the-art algorithms. Comparison studies present the newly developed algorithm's acceptability for handling the entire Fruit-360 dataset and achieving high computational efficiency. Furthermore, the Cascaded-ANFIS was employed in several other real-world applications in different fields, such as hydropower and flood and forecasting.
Hydropower is a crucial power source in the current world, and there is a vast range of benefits in forecasting power generation for the future. This study focuses on the significance of climate change on the future representation of the Samanalawewa Reservoir Hydropower Project in Sri Lanka using an architecture of the Cascaded ANFIS algorithm.
Moreover, the study assessed the capacity of the novel Cascaded ANFIS algorithm for handling regression problems and compared the results with the state-of-art regression models. The inputs to this system were the rainfall data of selected weather stations inside the catchment. The future rainfalls were generated using Global Climate Models at RCP4.5 and RCP8.5 and corrected for their biases. The Cascaded ANFIS algorithm was selected to handle this regression problem by comparing the best algorithm among the state-of-the-art regression models, such as RNN, LSTM, and GRU. The Cascaded ANFIS could forecast the power generation with a minimum error of 1.01, whereas the second-best algorithm, GRU, scored a 6.5 error rate. The predictions were carried out for the near future and mid-future and compared against the previous work. The results show that the algorithm can predict the variation of power generation with rainfall with a slight error rate. It was found that the research can be utilized in numerous areas for hydropower development.
The Rainfall-Runoff (R-R) relationship is essential to the hydrological cycle. Sophisticated hydrological models can accurately investigate R-R relationships; however, they require many data. Therefore, machine learning and soft computing techniques have taken the attention in the environment of limited hydrological, meteorological, and geological data. The accuracy of such models depends on the various parameters, including the quality of inputs and outputs and the used algorithms. However, identifying a perfect algorithm is still challenging. This study presents a method using Cascaded-ANFIS to predict runoff based on rainfall accurately. This study compared the model against three regression algorithms: Long Short-Term Memory, Grated Recurrent Unit, and Recurrent Neural Networks. These algorithms have been selected due to their outstanding performances in similar studies. The models were tested on the Mahaweli River, the

longest in Sri Lanka. The results showcase that the Cascaded-ANFIS-based model outperforms the other algorithms. In addition, Shared Socio-economic Pathways (SSP2-4.5 and SSP5-8.5 scenarios) were used to generate future rainfalls, forecast the near-future and mid-future water levels, and identify potential flood events. The future forecasting results indicate a decrease in flood events and magnitudes in both SSP2-4.5 and SSP5-8.5 scenarios. Furthermore, the SSP5-8.5 scenario shows drought weather from May to August yearly. The results of this study can effectively be used to manage and control water resources and mitigate flood damages.

Hydrologic models require atmospheric, dynamic and static models to simulate river flow from catchments. Thus the accuracy of hydrologic modelling highly depends on the data quality. Therefore, simulation is always challenging in data-scarcity environments. In addition, physical flow measurements are infeasible in the Spatiotemporal domain, and soft computing techniques are helpful in river flow simulation in data-scarcity environments. This research proposes model implementation using the Cascaded-ANFIS algorithm to simulate river flows in the Spatiotemporal domain with limited input rainfall data. The developed generic rainfall-river flow model was applied to five river basins in different geographical and climatic areas and tested its accuracy against the measured river flows. Six statical metrics were used to evaluate the accuracy of the developed model. In addition, the proposed algorithm was tested against state-of-the-art machine learning algorithms such as Recurrent Neural Network, Linear Regression, Ridge Regression, Lasso Regression, Long Short-Term Memory, and Gated Recurrent Units. Excellent acceptability of simulated river flows against measured river flows can be presented irrespective of land use, geographic, and climatic regions.

Moreover, hydrologic models to simulate river flows are computationally costly. In addition to the precipitation and other meteorological time series, catchment characteristics, including soil data, land use and land cover, and roughness, are essential in most hydrologic models. The unavailability of these data series challenged the accuracy of simulations, thus impacting related designs. However, recent advances in soft computing techniques offer better approaches and solutions at less computational complexity. These require a minimum amount of data; however, they reach higher accuracies depending on the quality of data sets. The Gradient Boosting Algorithms and Adaptive Network-based Fuzzy Inference System are two such systems that can be used in simulating river flows based on the catchment rainfall. In this paper, the computational capabilities of these two systems were tested in simulated river flows by developing the prediction models for Malwathu Oya in Sri Lanka. The simulated flows were then compared with the ground-measured river flows for accuracy. Correlation of coefficient (R), Per cent-Bias (bias), Nash Sutcliffe Model efficiency coefficient (NSE), Mean Absolute Relative Error (MARE), Kling-Gupta Efficiency (KGE), and Root mean square error (RMSE) were used as the comparative indices between Gradient Boosting Algorithms

and Adaptive Network-based Fuzzy Inference Systems. Results of the study showcased that both systems can simulate river flows as a function of catchment rainfalls; however, Cat gradient Boosting algorithm (CatBoost) has a computational edge over the Adaptive Network Based Fuzzy Inference System (ANFIS). The CatBoost algorithm outperformed other algorithms used in this study. However, more applications should be considered for sound conclusions. Therefore, as a conclusion of this dissertation thesis, it can be stated that the Cascaded-ANFIS has more outstanding capabilities and higher robustness than most of the traditional and black-box algorithms.

Acknowledgements

First of all, I would like to thank Professor Yukinobu Hoshino for his willingness to coach me, his promptness, the eye-opening discussions we had and his valuable comments on the progress of my research. In the same light, I thank my co-Supervisors, Dr Tuan Linh Dang, Professor Upaka Rathnayake and Professor Akira Miyazaki and for their excellent observations, ideas, and creativity that inspired my dissertation. I would like to thank all the professors at the Kochi University of Technology responsible for the PhD courses I have attended during these years. I have acquired considerable knowledge through their presentations and interactive scientific discussions. Thank you also, my laboratory mates, in the Hoshino Laboratory at the Kochi University of Technology, for creating the best working conditions. I would also like to thank all my other friends at the Kochi University of Technology for their friendship and support over the past three years. To all of them, I would like to express my sincere greetings.

Additionally, I would like to thank all office staff of the International Relations Division, Kochi University of Technology. This would not have been possible without my mother, father and brothers. Hence I thank them wholeheartedly. Finally, countless thanks and acknowledgements go to my wife for giving me her caring support and understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Machine Intelligence (MI), Artificial Intelligence (AI), and Machine Learning (ML) are not the same, but they are distinct from each other. Figuring out the difference in these terms is essential when implementing systems. For example, if the difference among the above three terms is unknown, the research outcome can become an automation project. Knowing the potential of MI, AI, and ML is essential. A machine can work as an automation system or as an intelligent system. Automation is simply the work done successfully by a computer, which humans assign. When a machine is programmed to learn trends and patterns, selecting the best option from a collection of possible options is ML. Machine intelligence is a crucial factor in the machine learning process. With the ability of MI, the machine becomes capable of tackling more complex problems Predicting severe retinopathy of prematurely using machine learning in healthcare [1], underwriting process algorithms [2], analysis in stock market [3] are some of the complex scenarios. AI can retain all data available today and generate brand-new solutions for challenges. AI has become superior to the human brain by producing solutions humans have never considered.

When an algorithm can approximate a solution to a problem, it is called a heuristic. Heuristic algorithms always depend on the problem environment. Hence, it is convenient to say that heuristic algorithms are problem-dependent. Nevertheless, there are methods/algorithms which can be used in many situations called meta-heuristic algorithms. These methods are independent of the problem environment, enabling the solution to be used in many problems. However, meta-heuristic is a descendant of heuristic algorithms [4]. Besides, a higher level of heuristic is meta-heuristic. Because meta-heuristic algorithms can select the best solutions by learning from past solutions and performing sophisticated search moves, meta-heuristic is not a trial and error method [5].

Meta-heuristic shines much better in Computational Intelligence (CI). For many years, CI has been an active topic in almost every field of science [6]. CI has recently begun to dominate in many fields by combining the power of meta-heuristic algorithms. [7]. Today, AI is capable enough to care for knowledge-based systems [8–11]. It is also proven that rather than using proper methods such as Neural Networks (NN) or Fuzzy Logic (FL), combining the techniques gives remarkable results in data-driven approaches such as Adaptive Network-Based Fuzzy Inference System (ANFIS) [12–15]. Therefore, recent researchers have been deploying this method in many fields such as control, identification, and prediction [16–18, 18–20]. The best trade-off between NN and FL is ANFIS. Jang, in 1993 introduced ANFIS to the world of artificial intelligence [21].

ANFIS is known to have a high degree of accuracy. Hence, it is used in many fields such as Engineering, medicine, transportation, business, and economics [22]. Many researchers found that ANFIS performs well when there are few inputs. In [23, 24], the authors have used less than five inputs to their ANFIS systems, and authors in [25] have used six inputs for their experiments. In recent years most researchers have tried to expand the ANFIS models to use more input data since this is the era of the Big-Data paradigm.

In recent years, a significant contribution to overcome the ANFIS limitations has been made by researchers worldwide. Reducing input data by selecting the best from the input set and reducing the rule base to overcome the computational complexity can be considered. Some novel implementations of ANFIS methods involve removing layers from the original ANFIS structure. In paper [26], the authors have removed the third layer to save computation time and reduce the complexity. Rather than using the raw version of ANFIS models, researchers tend to change the parameters and system variables to find better solutions in complex systems. Although knowledge-based systems can predict, control, and identify complex systems or models, combining optimization algorithms can result in better global minimum and computational complexity solutions.

Optimization is one of the primary processes in data learning techniques. The art of making the right decisions is called optimization. Hence, one of the most compelling tasks when dealing with many scientific and engineering problems is optimization [27].

Optimization can minimize costs and maximize efficiency. Generally, these tasks can be solved using traditional analytical methods. When the task is multi-model, multidimensional, and noisy, the general optimization methods find it hard to give the optimum solution. This problem results in more complex methods being born into the family of optimization. Many researchers in the world provide better solutions day by day. Nevertheless, the optimum solution is still out there to be found. Mostly, all the optimization algorithms are meta-heuristic algorithms. Some existing optimization algorithms which give immediate solutions to the optimum are Genetic Algorithms (GA) [28–30], Particle Swarm Optimization (PSO) [31–34], Artificial Bee Colony (ABC) [35–37], Ant Colony Optimization (ACO) [38] and Bacterial Foraging (BF) [39]. In [40], Cat Swarm Optimization (CSO) is occupied with sharpening ANFIS. It can be observed that these optimization methods were inspired by nature.

There is also a considerable effort in overcoming the limitation of the curse of dimensionality of ANFIS. In [41], a wrapper feature selection method is introduced for the Twitter sentiment classification model. With a big data processing platform, a feature selection method is presented in the paper [42]. Authors in [43] discuss a novel intuitionistic fuzzy clustering algorithm for multiple object tracking based on feature selection. A feature selection that uses a fuzzy boundary area for the nearest neighbour classification is presented in [44].

In recent times, fuzzy logic has been extensively used due to nonlinear properties, robustness, and easy implementation [45]. Some research has been performed to reconstruct the input features using fuzzy sets to reduce the influence of uncertainties [46]. Nonetheless, there is still a qualitative difference between the desired and experimental outputs. Previous research has shown four drawbacks of the existing algorithms as follows [47].

1. Computational burden: It is a common problem in all optimization problems. Though the algorithms have high accuracy, computational power is more difficult to obtain.

2. Time consumption: Time consumption has also become a huge obstacle researchers must overcome.

3. Application restriction: The state-of-the-art algorithms are not feasible enough to use in every application due to the lack of adaptiveness.

4. Curse of dimensionality: There is always a typical drawback when fuzzy logic is engaged in an algorithm known as the curse of dimensionality. Typically, using fuzzy logic is challenging to handle multidimensional data as inputs. Hence, most algorithms use the pre-processing stage, where they use feature selection steps to determine the best features and reduce the dimension of the inputs.

1.2 Research objectives

According to the above analysis, the novel approach called Cascaded ANFIS is designed to overcome most problems. Consequently;

1. presents a novel optimization algorithm based on ANFIS for classification problems. This method provides computational simplicity due to using two input ANFIS models as the base.

2. capability of handling a vast amount of input variables. Data reduction is not compulsory for multi-variable data sets. Since the noise data is used in training the model, implementing this algorithm is promising in real time.

3. Instead of determining the data redundancy, this algorithm presents a novel pair selection method using the same two-input ANFIS model.

Moreover, the proposed algorithm was employed in many applications to bring the most accurate solutions when compared with the state-of-the-art traditional and black box algorithms. The real-time applications where the Cascaded ANFIS was employed can be pointed out as follows.

1. Image classification task

2. Hydropower forecasting

3. Rainfall-runoff prediction and forecasting

4. Germination of seeds prediction

1.3 Structure of the dissertation

This dissertation is divided into eight chapters, each dealing with different accounts. The content of the dissertation is as follows.

- Chapter 2: Introduction proof and implementation of the novel algorithm: The Cascaded ANFIS – A novel Adaptive Network-based Fuzzy Inference System Approach.
- Chapter 3: Application of Image classification - Fruit 360 dataset Classification.
- Chapter 4: Application of Energy generation Prediction and Forecasting - Hydropower forecasting in Sri Lanka.
- Chapter 5: Application of Flood Forecasting - Mahaweli river flood prediction, a case study in Sri Lanka.
- Chapter 6: Application of Generic rainfall-runoff model implementation - A combined case study from Japan, Vietnam and Sri Lanka.
- Chapter 7: Investigation of computational capability of Gradient Boosting algorithms - Investigation of Malwathu river flooding, a case study in Sri Lanka.
- Chapter 8: Investigation of Seeds classification - Rice seeds classification based on age.
- Chapter 9: Designing and Simulation of ANFIS-based UAV controller.
- Chapter 10: Investigation of Hand gesture recognition - Effective attempt to recognize hand gestures using gradient boosting algorithms.

References

[1] Tamer Karatekin, Selim Sancak, Gokhan Celik, Sevilay Topcuoglu, Guner Karatekin, Pinar Kirci, and Ali Okatan. Interpretable machine learning in healthcare through generalized additive model with pairwise interactions (ga2m): Predicting severe retinopathy of prematurity. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pages 61–66. IEEE, 2019.

[2] Yi Tan and Guo-Ji Zhang. The application of machine learning algorithm in underwriting process. In *2005 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3523–3527. IEEE, 2005.

[3] Kunal Pahwa and Neha Agarwal. Stock market analysis using supervised machine learning. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 197–200. IEEE, 2019.

[4] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.

[5] Xin-She Yang, Su Fong Chien, and Tiew On Ting. Computational intelligence and metaheuristic algorithms with applications, 2014.

[6] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[7] Amir Hossein Gandomi, Siamak Talatahari, Xin-She Yang, and Suash Deb. Design optimization of truss structures using cuckoo search algorithm. *The Structural Design of Tall and Special Buildings*, 22(17):1330–1349, 2013.

[8] Bin Wang, Jianyi Xue, and Diyi Chen. Takagi-sugeno fuzzy control for a wide class of fractional-order chaotic systems with uncertain parameters via linear matrix inequality. *Journal of Vibration and Control*, 22(10):2356–2369, 2016.

[9] Sy Dzung Nguyen, Quoc Hung Nguyen, and Seung-Bok Choi. Hybrid clustering based fuzzy structure for vibration control–part 1: A novel algorithm for building neuro-fuzzy system. *Mechanical Systems and Signal Processing*, 50:510–525, 2015.

[10] RM Namal Bandara and Sujeetha Gaspe. Fuzzy logic controller design for an unmanned aerial vehicle. In *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–5. IEEE, 2016.

[11] RMNB Ratnayake, TS De Silva, and CJ Rodrigo. A comparison of fuzzy logic controller and pid controller for differential drive wall-following mobile robot. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 523–528. IEEE, 2019.

[12] Massimo Panella and Antonio Stanislao Gallo. An input-output clustering approach to the synthesis of anfis networks. *IEEE Transactions on fuzzy systems*, 13(1):69–81, 2005.

[13] Sy Dzung Nguyen, Bao Danh Lam, Quoc Hung Nguyen, and Seung-Bok Choi. A fuzzy-based dynamic inversion controller with application to vibration control of

vehicle suspension system subjected to uncertainties. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 232 (9):1103–1119, 2018.

[14] Sy Dzung Nguyen, Quoc Hung Nguyen, and Tae-Il Seo. Anfis deriving from jointed input-output data space and applying in smart-damper identification. *Applied Soft Computing*, 53:45–60, 2017.

[15] Hamed Khodadadi and Hamid Ghadiri. Fuzzy logic self-tuning pid controller design for ball mill grinding circuits using an improved disturbance observer. *Mining, Metallurgy & Exploration*, 36(6):1075–1090, 2019.

[16] Hui Li, Chen Guo, Simon X Yang, and Hongzhang Jin. Hybrid model of wt and anfis and its application on time series prediction of ship roll motion. In *The Proceedings of the Multiconference on" Computational Engineering in Systems Applications"*, volume 1, pages 333–337. IEEE, 2006.

[17] De-Wang Chen and Jun-Ping Zhang. Time series prediction based on ensemble anfis. In *2005 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3552–3556. IEEE, 2005.

[18] Abduh Muhammad, Yul Y Nazaruddin, and Parsaulian I Siregar. Design of non-linear adaptive-predictive control system with anfis modeling for urea plant reactor unit. In *2019 6th International Conference on Instrumentation, Control, and Automation (ICA)*, pages 147–152. IEEE, 2019.

[19] Mehrnoush Davanipour, Maryam Zekri, and Farid Sheikholeslam. The preference of fuzzy wavelet neural network to anfis in identification of nonlinear dynamic plants with fast local variation. In *2010 18th Iranian Conference on Electrical Engineering*, pages 605–609. IEEE, 2010.

[20] JKEC de Martins and FMU de Araújo. Nonlinear system identification based on modified anfis. In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Colmar, France*, pages 21–23, 2015.

[21] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.

[22] Samarjit Kar, Sujit Das, and Pijush Kanti Ghosh. Applications of neuro fuzzy systems: A brief review and future outline. *Applied Soft Computing*, 15:243–259, 2014.

[23] Ali Barati-Harooni, Adel Najafi-Marghmaleki, and Amir H Mohammadi. Anfis modeling of ionic liquids densities. *Journal of Molecular Liquids*, 224:965–975, 2016.

[24] Afshin Tatar, Ali Barati-Harooni, Adel Najafi-Marghmaleki, Behzad Norouzi-Farimani, and Amir H Mohammadi. Predictive model based on anfis for estimation of thermal conductivity of carbon dioxide. *Journal of Molecular Liquids*, 224:1266–1274, 2016.

[25] Osman Taylan and Bahattin Karagözoğlu. An adaptive neuro-fuzzy model for prediction of student's academic performance. *Computers & Industrial Engineering*, 57(3):732–741, 2009.

[26] Alireza Peymanfar, Abdollah Khoei, and Kh Hadidi. A new anfis based learning algorithm for cmos neuro-fuzzy controllers. In *2007 14th IEEE International Conference on Electronics, Circuits and Systems*, pages 890–893. IEEE, 2007.

[27] Piotr Dziwiński and Łukasz Bartczuk. A new hybrid particle swarm optimization and genetic algorithm method controlled by fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 2019.

[28] Taoping Liu, Wentian Zhang, Peter McLean, Maiken Ueland, Shari L Forbes, and Steven W Su. Electronic nose-based odor classification using genetic algorithms and fuzzy support vector machines. *International Journal of Fuzzy Systems*, 20(4): 1309–1320, 2018.

[29] Harihar Kalia, Satchidananda Dehuri, Ashish Ghosh, and Sung-Bae Cho. Surrogate-assisted multi-objective genetic algorithms for fuzzy rule-based classification. *International Journal of Fuzzy Systems*, 20(6):1938–1955, 2018.

[30] Chuanchao Zhang. Classification rule mining algorithm combining intuitionistic fuzzy rough sets and genetic algorithm.

[31] Tien-Loc Le, Tuan-Tu Huynh, and Chih-Min Lin. Self-evolving interval type-2 wavelet cerebellar model articulation control design for uncertain nonlinear systems using pso. *International Journal of Fuzzy Systems*, 21(8):2524–2541, 2019.

[32] MH Fazel Zarandi and F Dorry. A hybrid fuzzy pso algorithm for solving steelmaking-continuous casting scheduling problem. *International Journal of Fuzzy Systems*, 20(1):219–235, 2018.

[33] Chih-Min Lin and Tien-Loc Le. Pso-self-organizing interval type-2 fuzzy neural network for antilock braking systems. *International Journal of Fuzzy Systems*, 19 (5):1362–1374, 2017.

[34] Chi-Chung Chen. Optimization of zero-order tsk-type fuzzy system using enhanced particle swarm optimizer with dynamic mutation and special initialization. *International Journal of Fuzzy Systems*, 20(5):1685–1698, 2018.

[35] Ehsan Zakeri, Seyed Alireza Moezi, and Mohammad Eghtesad. Tracking control of ball on sphere system using tuned fuzzy sliding mode controller based on artificial bee colony algorithm. *International Journal of Fuzzy Systems*, 20(1):295–308, 2018.

[36] E Sasikala and N Rengarajan. An intelligent technique to detect jamming attack in wireless sensor networks (wsns). *International Journal of Fuzzy Systems*, 17(1): 76–83, 2015.

[37] Chung-Hsien Kuo, Yu-Cheng Kuo, Hung-Chyun Chou, and Yi-Tseng Lin. P300-based brain–computer interface with latency estimation using abc-based interval type-2 fuzzy logic system. *International Journal of Fuzzy Systems*, 19(2):529–541, 2017.

[38] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[39] Z Zakaria, TKA Rahman, and EE Hassan. Economic load dispatch via an improved bacterial foraging optimization. In *2014 IEEE 8th International Power Engineering and Optimization Conference (PEOCO2014)*, pages 380–385. IEEE, 2014.

[40] Meysam Orouskhani, Mohammad Mansouri, Yasin Orouskhani, and Mohammad Teshnehlab. A hybrid method of modified cat swarm optimization and gradient

descent algorithm for training anfis. *International Journal of Computational Intelligence and Applications*, 12(02):1350007, 2013.

[41] NK Suchetha, Anupama Nikhil, and P Hrudya. Comparing the wrapper feature selection evaluators on twitter sentiment classification. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–6. IEEE, 2019.

[42] Mehmet Burak Çatalkaya, Oya Kalıpsız, Mehmet S Aktaş, and Umut Orçun Turgut. Data feature selection methods on distributed big data processing platforms. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 133–138. IEEE, 2018.

[43] Liang-qun Li, Xiao-li Wang, Zong-xiang Liu, and Wei-xin Xie. A novel intuitionistic fuzzy clustering algorithm based on feature selection for multiple object tracking. *International Journal of Fuzzy Systems*, 21(5):1613–1628, 2019.

[44] Tae-Chon Ahn, Seok-Beom Roh, Yong Soo Kim, and Jihong Wang. Prototypes reduction and feature selection based on fuzzy boundary area for nearest neighbor classifiers. *International Journal of Fuzzy Systems*, 21(2):639–654, 2019.

[45] Hong-Gui Han, Zhi-Yuan Chen, Hong-Xu Liu, and Jun-Fei Qiao. A self-organizing interval type-2 fuzzy-neural-network for modeling nonlinear systems. *Neurocomputing*, 290:196–207, 2018.

[46] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2016.

[47] Mohd Najib Mohd Salleh, Noureen Talpur, and Kashif Hussain. Adaptive neuro-fuzzy inference system: Overview, strengths, limitations, and solutions. In *International Conference on Data Mining and Big Data*, pages 527–535. Springer, 2017.

Chapter 2

The Cascaded ANFIS – A novel Adaptive Network-based Fuzzy Inference System Approach

Chapter 2 presents a comprehensive explanation of the related work, the theory behind the algorithm, and the implementation flow of the proposed algorithm. This work is published in the International Journal of Fuzzy systems [1].

2.1 Related Works

2.1.1 Non-Linear optimization

Searching for the largest or smallest value with given constraints is called nonlinear optimization within a nonlinear objective function. Consider the following minimization problems. The minimization nonlinear optimization problem can be defined as in equation (1);

$$\text{Minimize} \qquad f(x)$$

$$\begin{aligned}
&\text{subjected to:} \\
&g_i(X) \leq 0, \qquad && (i = 1, 2, 3, ..., p) \\
&h_j(X) = 0, \qquad && (j = 1, 2, 3, ..., q) \\
&x_n \in |\underline{x^r}, \overline{x^r}|, \qquad && (d = 1, 2, 3, ..., r)
\end{aligned} \tag{2.1}$$

Here, the optimization function is f where $X = [x^1, ..., x^r]^T$ is the solution. The maximum and minimum permissible are $\overline{x^r}$ and $\underline{x^r}$. Moreover, the optimization function is

r dimensional, and equality and inequality constraints are given by $g_i(X)$ and $h_j(X)$, respectively. At a glance, there can be different methods of reaching the solution space for this specific optimization problem, such as PSO, GA, ABC, and ACO. Most of these methods are population-based algorithms.

2.1.2 ANFIS algorithm

Jang introduced a versatile and very intelligent hybrid system called ANFIS in 1993. ANFIS is a perfect collaboration between neural networks (NN) and fuzzy inference systems (FIS) [2]. The collaboration of NN and FIS brings their strengths to the ANFIS system. Moreover, the most significant advantage of this network is the system's transformation to simple if-then rules [3]. The arrangement of if-then rules of ANFIS provides the ability to deal with non-linear functions. It is proven that ANFIS has been used in many research areas and shows powerful results overall. ANFIS is well known to combine with various ranges of algorithms to decrease training phase error. For example, gradient descent and the least square method can be combined and used to optimise the effectiveness of searching for the best parameters.

ANFIS works similarly to the fuzzy system, which Takagi introduced, to Sugeno in 1985 [4]. A least-squares approach is used to determine the consequence factors in the forward section, and the backward learning phase is based on the gradient least-squares approach. Then, gradient descent in the regressive advance is used to reset the parameters.

Generally, ANFIS consists of five layers: the input layer, the membership function layer, the fuzzification layer, the defuzzification layer, the normalization layer, and the output layer, respectively. Further explanation is based on Figure 2.1, assuming two inputs into the ANFIS system, namely x and y. The output is f. Following the Sugeno FIS, the if-then rule configuration of ANFIS can be denoted in the following equation.

$$f_1 = p_1 x + q_1 y + r_1, \qquad \text{assume} \quad x = A_1, \quad y = B_1$$

$$f_2 = p_2 x + q_2 y + r_2, \qquad \text{assume} \quad x = A_2, \quad y = B_2$$

Here, A_1 and B_1 are fuzzy sets, and p_i, q_i, and r_i are design parameters where $i = 1, 2$. The first layer of the ANFIS structure is the membership layer. All the nodes in this layer are adaptive. Membership grades are generated in this layer for each input. The functionality can be expressed as in the following equations:

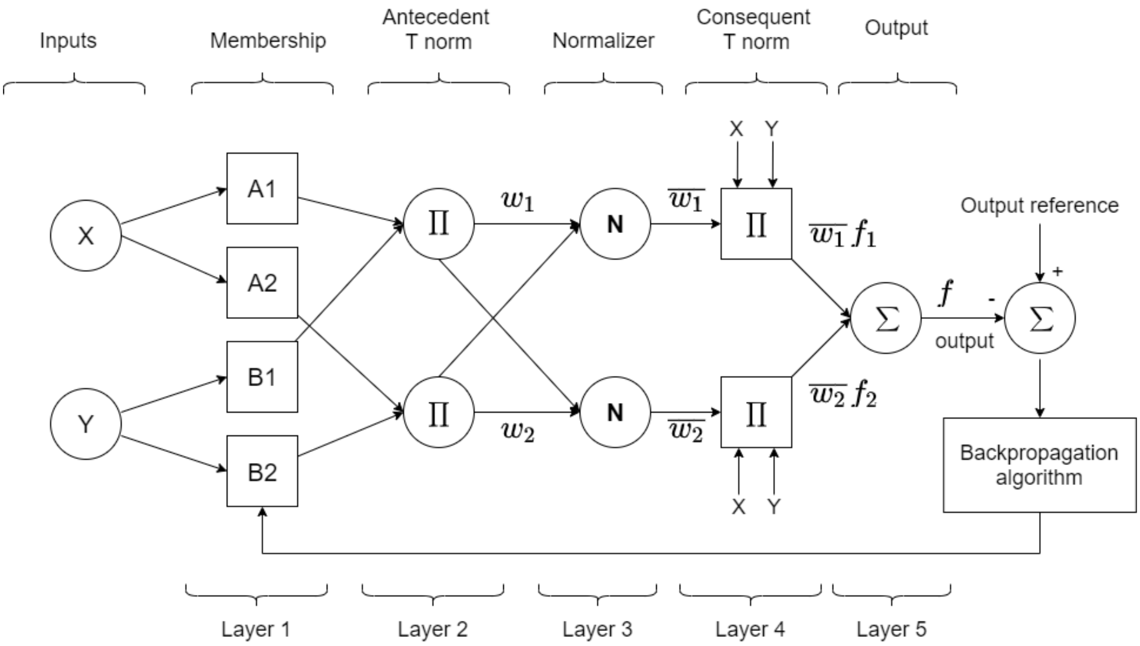


Figure 2.1: ANFIS schematic view

$$O_{1,i} = \mu_{Ai}(x) \qquad i = 1, 2 \tag{2.2}$$

$$O_{1,j} = \mu_{Bj}(y) \qquad j = 1, 2 \tag{2.3}$$

Where x and y are the inputs. Linguistic labels for the nodes are denoted as Ai and Bi. $\mu_{Ai}(x)$, and $\mu_{Bj}(y)$ are adaptable, and they are the membership grades for a fuzzy set A ($A1, A2, B1$ and $B2$). For instance, if the membership functions are bell-shaped, the following equation is employed.

$$\mu_{Ai}(x) = \frac{1}{1 + \left\{ (\frac{x - c_i}{a_i})^2 \right\}^{b_i}} \tag{2.4}$$

Here, bell-shaped function parameters are a_i, b_i, and c_i accordingly.
In the next layer, simple multiplication is performed, and this layer consists of fixed nodes. The mathematical expression of the layer can be presented as follows.

$$O_{2,i} = w_i = \mu_{Ai}(x) \times \mu_{Bi}(x) \qquad i = 1, 2 \tag{2.5}$$

The next layer is a fixed node normalization layer. The normalization of the output from the second layer is performed in this layer. The following equation shows the operation.

$$O_{3,i} = \overline{w} = \frac{w_i}{w_1 + w_2} \qquad i = 1, 2 \tag{2.6}$$

Here, the firing strength of node$i$ is presented by w_i.

The fourth layer can simplify the production of the normalized output from the third layer. This layer is adaptive, and the output can be presented using the following equation.

$$O_{4,i} = \overline{w} f_i = \overline{w_i}(p_i + q_i + r_i) \qquad i = 1, 2 \tag{2.7}$$

The final layer has only one node, and it is fixed. This node does the summation of all incoming inputs. Finally, the overall outcome can be prevented by using the following equation.

$$O_{5,i} = \sum_{i=1}^{2} \overline{w_i} f_i = \frac{w_i f_1 + w_2 f_2}{w_1 + w_2} \tag{2.8}$$

ANFIS has better learning ability because back-propagation and least square approaches make the system more precise and faster convergence, as mentioned before; there are six consequent parameters in this system (assuming bell-shaped membership functions are used). To obtain the best cost, tuning these parameters is the main objective of this ANFIS system. Back-propagation is determined to change the parameters in the first layer, and the least square approach is responsible for the fourth layer parameter tuning [5].

2.2 The proposed algorithm - Cascaded ANFIS

As mentioned in the above paragraphs, ANFIS has its limitations. Mainly, the curse of dimensionality and the computational complexity [6]. Researchers propose some methodologies to overcome this cause, but the final result is questionable. Hence, this novel optimization algorithm aims to narrow the solution space between prediction and reality.

In this section, the novel Cascaded ANFIS method is presented in detail. The overall algorithm can be introduced using Figure 2.2. This algorithm can also be introduced as an extension of ANFIS. Because rather than having five layers for the ANFIS algorithm, there are iterations that can navigate the solution to be more precise.

Compared to the novel algorithm with the traditional ANFIS algorithm, the main difference is that the output of the traditional ANFIS algorithm becomes the input of the subsequent usage of the traditional ANFIS algorithm. But, as in the general ANFIS algorithm, fuzzy is used for the fuzzification process in the inner layers of the ANFIS model. Fuzzification is performed using membership functions by converting numerical values into fuzzy members.
The Cascaded ANFIS algorithm consists of two main modules.

1. Pair Selection Module

2. Training Module

The Pair Selection module gives a solution to the first main limitation of ANFIS. It is general practice to reduce the input features before using an algorithm. But the novel algorithm uses all the features to build a robust model, which can also benefit noisy data sets. The Training Module of the novel Cascaded ANFIS algorithm handles computational complexity. Each step can be introduced in detail as follows.

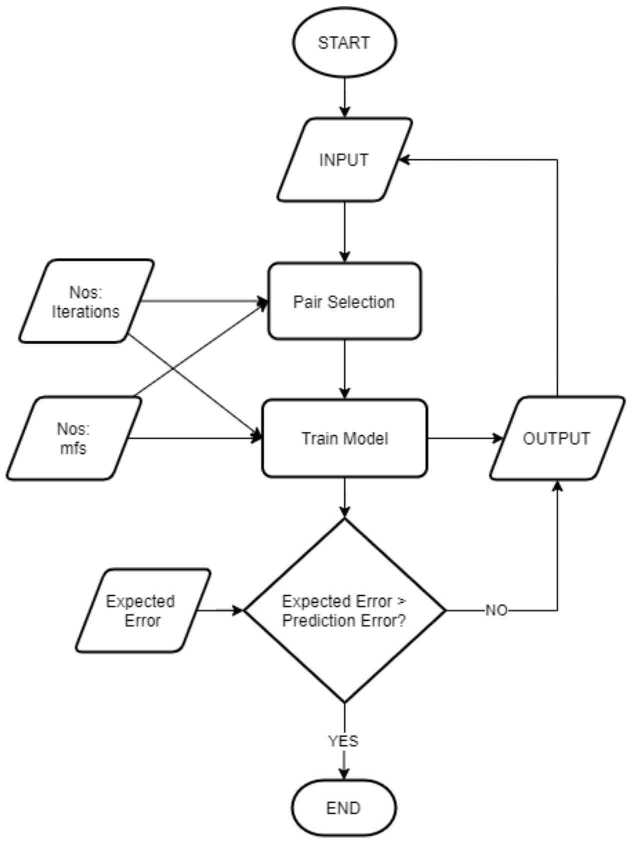


Figure 2.2: Overall flow diagram of novel Cascaded ANFIS algorithm

2.2.1 Input module

Here, the raw inputs are fed into the Cascaded ANFIS model. Initially, the inputs are paired using the Pair selection module. This particular ANFIS system uses a single module of two input ANFIS models to calculate all the solution points. The usage of the two-input ANFIS model is explained in the next section.

2.2.2 Pair selection module

The pair Selection module is a Sequential Feature Selection (SFS) process. The complete process of the pair selection module can be demonstrated using figure 2.3. The novelty of this method uses two inputs and one output ANFIS model to determine the best match for each input variable.

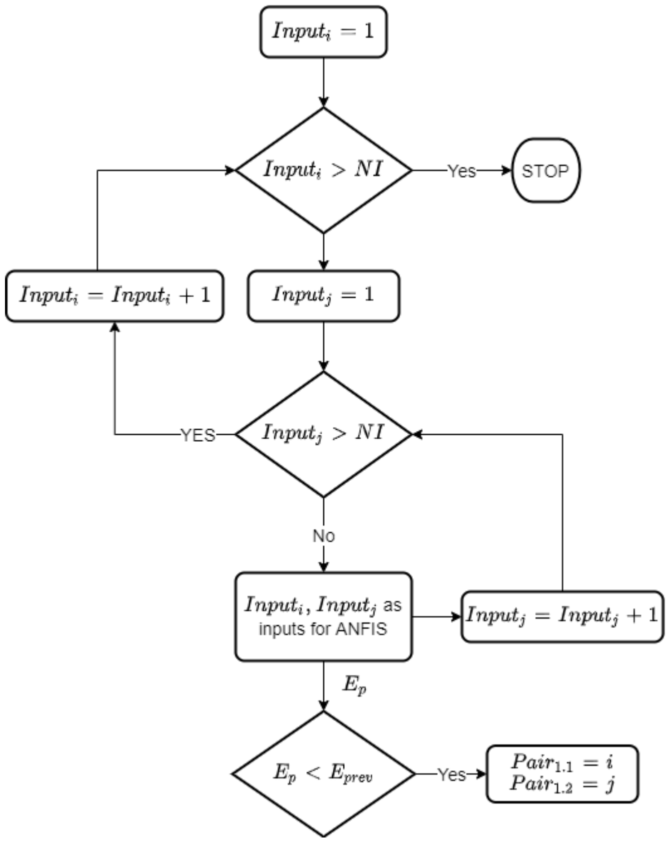


Figure 2.3: Pair Selection Module Structure

As shown in the figure, the final output is the matching pair. Therefore, a nested loop is used to go through every two pair combinations. In the figure, NI is the number of input variables. The first two input variables are initially selected and named $input_i$ and $input_j$. They are used as the input of the two-input ANFIS model, as shown in the figure. The root means square error (RMSE) is calculated and stored, and then the

RMSE (E_p) is checked against the previous RMSE (E_{prev}). At the end of the second loop, the matching pair can be extracted by observing the lowest RMSE value. Once the pairs are selected, the training phase can be initialized.

2.2.3 Train model module

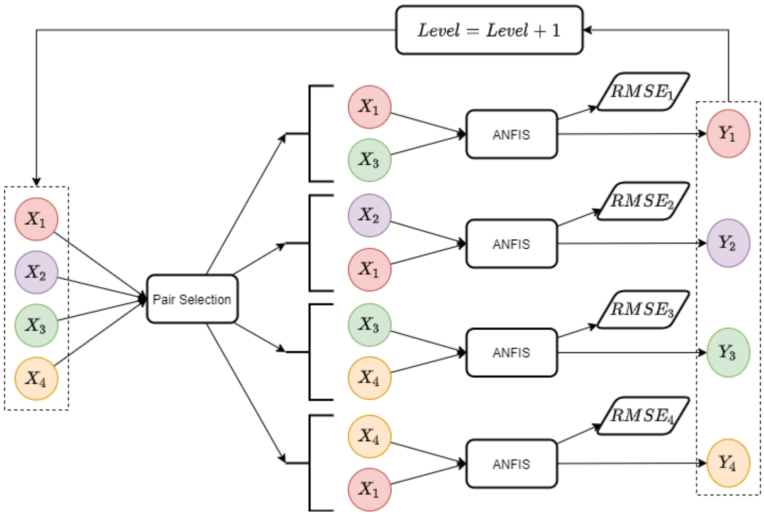


Figure 2.4: Example of Train Module Structure

Two input ANFIS model is adopted here as well. Since the input variables are paired with the best match from the previous module, the input can be delivered directly to the ANFIS module, generating current outputs and RMSE for each data pair. At this point, there is also a pre-defined target error. Thus, the RMSE is compared with the

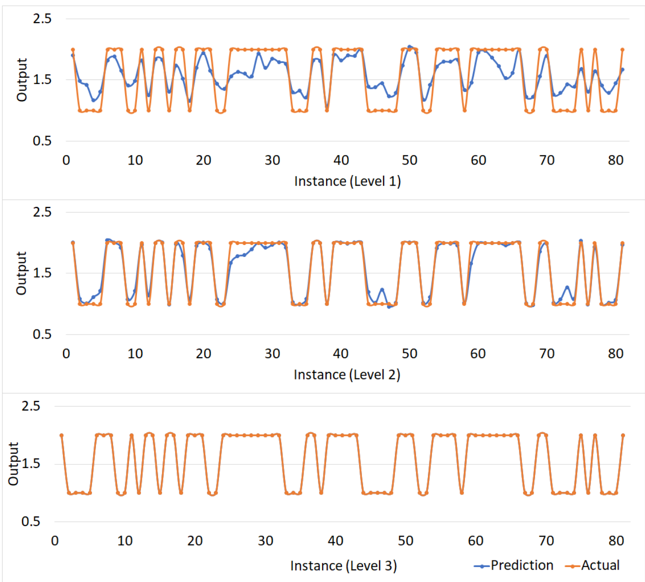


Figure 2.5: Performance for Breast Cancer data set

Table 2.1: ANFIS network parameters

Configuration
Membership Parameter
Kalman Parameters
Nodes
Number of membership functions
Number of inputs

Algorithm 1: Pseudo-code for Pair selection

```
initialization
MaxIterations = x
(data_input, data_output) = LoadData
ni = size(input variables)
while MaxIterations is not equal to 0 do
    if MaxIterations = 1 then
        input = data_input
    else
        input = output_prev
    end
    for i = 1:ni do
        for j = 1:ni do
            (network,output_prev,RMSE) = ANFIS(input(i), input(j),data_output)
            if RMSE is less than min_error save pair indexes
        end
        pair = pair + 1
    end
    Iterations = Iterations - 1
end
```

target error. If the target error is achieved, the process can be terminated. Else, the algorithm advances to the second iteration. The process of the iteration advancement can be explained in detail using figure 2.4.

Figure 2.4 illustrates an example approach of the Cascaded ANFIS model. As shown here, assume that there are four input variables in an optimization problem called X_1,X_2,X_3, and X_4, respectively.

$$input = \{X_1, X_2, X_3, X_4\} \tag{2.9}$$

As explained in the pair selection section, the input is paired with the best match as shown in equation 2.10 below.

$$input_{pairs} = \{X_1, X_3\}, \{X_2, X_1\}, \{X_3, X_4\}, \{X_4, X_1\} \tag{2.10}$$

Algorithm 2: Pseudo-code for Train model

```
initialization
MaxIterations = x
(data_input, data_output) = LoadData
ni = size(input variables)
Iterations = 1
while Iteration is not equal to MaxIterations do
    if Iterations = 1 then
        input = data_input
    else
        input = output_prev
    end
    for i = 1:ni do
        (network,output_prev,RMSE) = ANFIS(input(pair_1), input(pair_2),data_output)
        nets [Iterations][i] = network
        output[i] = output_prev
        error[Iterations][i] = RMSE
    end
    Iterations = Iterations + 1
end
```

Algorithm 3: Pseudo-code for the testing process

```
initialization
MaxIterations = x
Load data from training process
ni = size(input variables)
Iterations = 1
while Iteration is not equal MaxIterations do
    if Iterations != 1 then
        input = output_prev
    end
    for i = 1:ni do
        import ANFIS network
        import pair combination
        output = EVALUATE (network, pair combination)
        Calculate the best cost
    end
    Iterations = Iterations + 1
end
```

Then, using two input ANFIS models for each pair, two outputs are generated, namely, $RMSE_i$ and the predicted output (Y_i). They can be obtained using the following equations 2.11 and 2.12.

$$RMSE = \sqrt{(A - P)^2}$$

$$RMSE_{A,P} = \left[\sum_{i=1}^{N} \frac{(O_{Ai} - O_{Pi})^2}{N} \right]^{\frac{1}{2}} \tag{2.11}$$

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 + \frac{w_3}{w_2 + w_3} f_3 + \frac{w_4}{w_3 + w_4} f_4 \tag{2.12}$$

Where A and P are actual results and predicted results, respectively. N is the sample size. Obtaining the results for RMSE and Y completes the initial iteration. The RMSE error can now be compared with the goal error and proceed to the next iteration accordingly. When moving to the next iteration, the speciality is that the output from the first iteration, which are Y_1, Y_2, Y_3, and Y_4, will act as inputs for the second iteration.
Note that the first iteration generated four unique ANFIS network parameter sets. The second iteration also generates four unique ANFIS network parameter sets. ANFIS parameters that are used in this implementation are stated in Table 1. These parameters are used in the testing section of the algorithm.
In each iteration, mainly, six parameters are adjusted to narrow the error between the prediction and the actual results in each ANFIS structure. As shown in figure 2.4, the example has four inputs. Hence, four ANIFS structures have been used to obtain the outputs on the first iteration. In this case, the number of parameters that have been tuned is 24. Because each ANFIS structure has p, q, and r design parameters and membership parameters (if bell-shaped: 3 parameters), the first iteration has 24 parameters ($6 * 4$ ANFIS structures) for tuning. If the system advances to the next iteration, again, there will be another 24 unique parameters. Hence, increasing the number of iterations increases the complexity of the novel ANFIS algorithm. But, the algorithm is designed to operate only one ANFIS structure simultaneously. Hence, the complexity stays stable as two inputs ANFIS structure. However, the number of tuned parameters increases the accuracy of the optimization solution. Breast Cancer data set optimization performance in each iteration is shown in Fig 2.5, which provides a better understanding of the algorithm.

2.2.4 Pseudo-code explanation

The pseudo-codes for each section is demonstrated in the below sections.
The above pseudo-code describes how the pair selection operates on the input variables. A two-input ANFIS model obtains the network variables, outputs, and RMSE values.

Nested loops are responsible for going through all possible combinations of input variables. After selecting the best pairs for the two input ANFIS model, the training process can be advanced.

Algorithm 2 presents the pseudo-code for the training model. As shown in the diagram, for each input variable, a unique ANFIS model is dedicated to providing the outputs such as networks, outputs, and RMSE. These outputs are stored later in the iteration shifting and training phase.

Testing is performed using the results of the training step. As presented in algorithm 3, the data is imported and used to evaluate the Cascaded ANFIS method. At the end of the testing process, the best cost is calculated using the predicted and actual results.

Table 2.2: Data sets used for evaluating the Cascaded-ANFIS

Dataset	Features	Classes	Instances
IRIS	4	3	150
Breast cancer	9	2	116
Statlog (Vehicle Silhouettes)	18	4	946
Sports articles for objectivity analysis	59	2	1000
Superconductivity	81	7	21263
Musk 1	168	2	6598
Human Activity Recognition Using Smartphones	561	6	10299

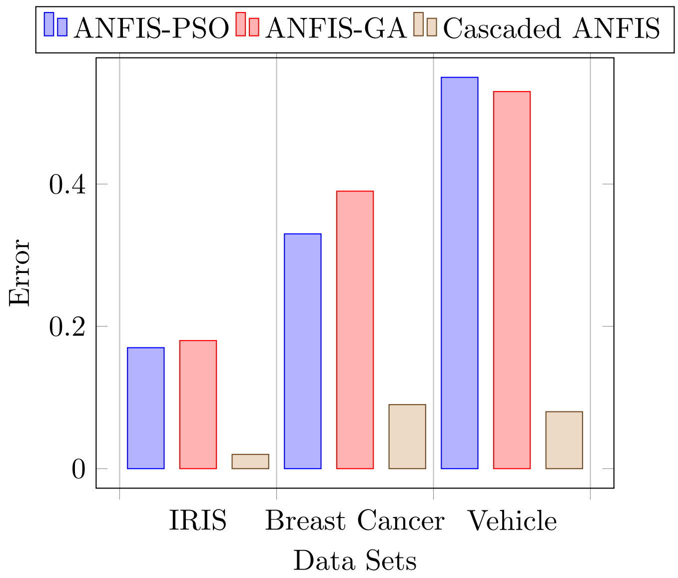


Figure 2.6: RMSE comparison of the Cascaded-ANFIS evaluation

Table 2.3: IRIS dataset for testing

	ANFIS-PSO	ANFIS-GA	Cascaded ANFIS
RMSE	0.181743	0.174594	0.005674368
MSE	0.033031	0.030483	3.21985E-05
MAE	0.132536	0.128472	0.005674
MAPE	7.825673	7.381097	0.245739
Correlation	0.977487	0.979315	0.999944

Table 2.4: Breast Cancer dataset for testing

	ANFIS-PSO	ANFIS-GA	Cascaded ANFIS
RMSE	0.497408	0.444374	0.086713
MSE	0.247415	0.197468	0.007519
MAE	0.360477	0.387072	0.050746
MAPE	25.47778	31.24708	3.937956
Correlation	0.557952	0.488011	0.987396

Table 2.5: Vehicle dataset for testing

	ANFIS-PSO	ANFIS-GA	Cascaded ANFIS
RMSE	0.653112	0.679	0.000324
MSE	0.426556	0.461041	1.05E-07
MAE	0.541745	0.533102	0.000229
MAPE	30.69937	25.08795	0.010369
Correlation	0.821025	0.788542	1

Table 2.6: Testing time comparison (seconds)

Dataset	ANFIS-GA	ANFIS-PSO	Cascaded-ANFIS
IRIS	0.184	0.399	0.1442
Breast	0.1824	0.4352	0.1156
Vehicle	0.384	0.8246	0.1697

Table 2.7: Training time comparison (seconds)

Dataset	ANFIS-GA	ANFIS-PSO	Cascaded-ANFIS
IRIS	1697.9	1479.2	743.4052
Breast	5259.807	4278.7	3314.84
Vehicle	15959.39	13055.64	10028.63

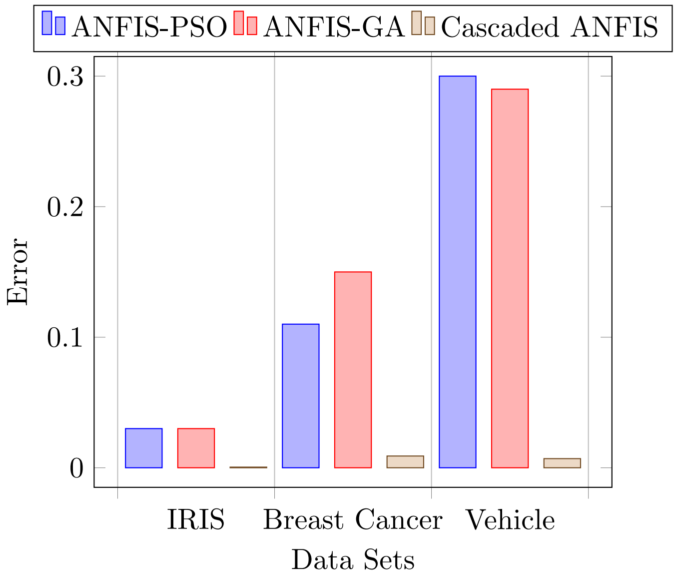


Figure 2.7: MSE comparison of the Cascaded-ANFIS algorithm evaluation

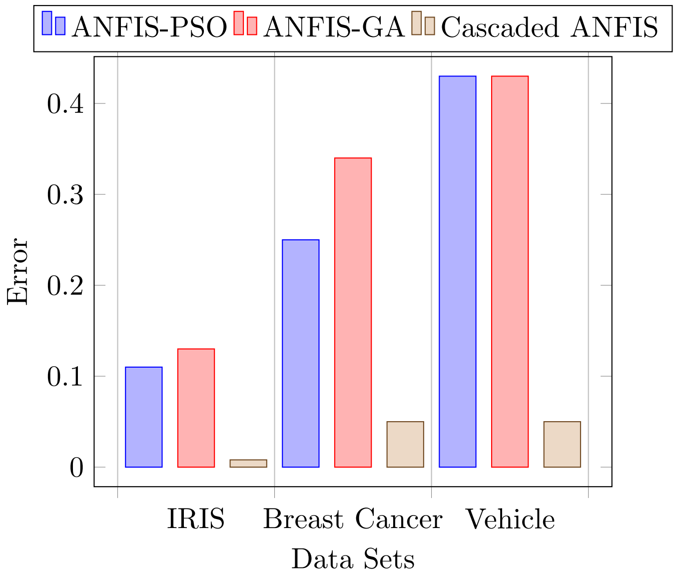


Figure 2.8: MAE comparison of the Cascaded-ANFIS algorithm evaluation

2.3 Experimental Design

2.3.1 Data sets

The research is conducted for seven publicly recognized data sets from the UCI machine learning repository [7] as follows:

1. IRIS
2. Breast Cancer
3. Statlog (Vehicle Silhouettes)

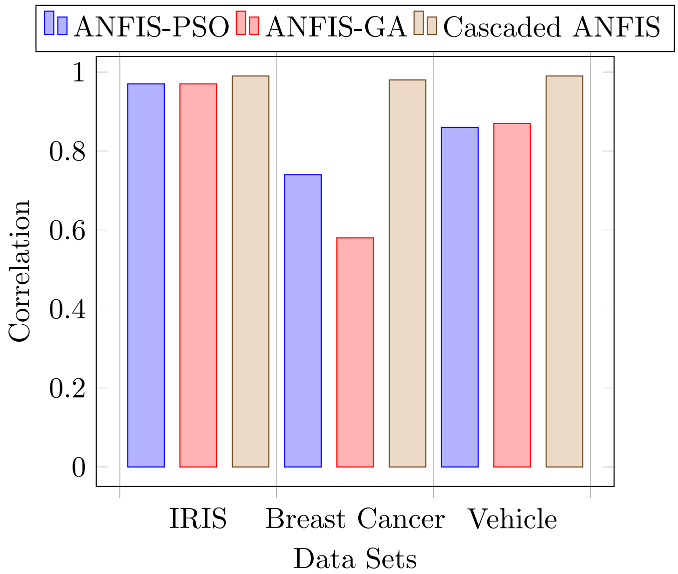


Figure 2.9: Correlation comparison of the Cascaded-ANFIS algorithm evaluation

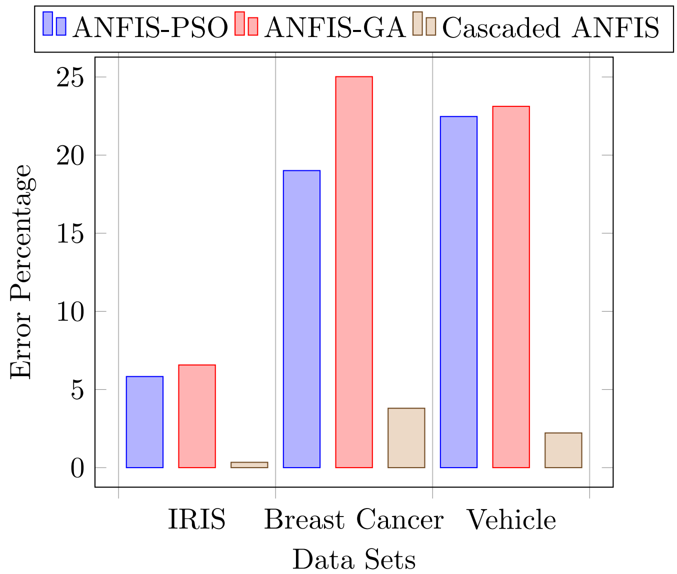


Figure 2.10: MAPE comparison of the Cascaded-ANFIS algorithm evaluation

4. Sports articles for objectivity analysis

5. Superconductivity

6. Musk 1

7. Human Activity Recognition Using Smartphones

Table 2 presents further details of each data set used. These data sets are different in several aspects, such as the number of classes, the number of input variables, and the

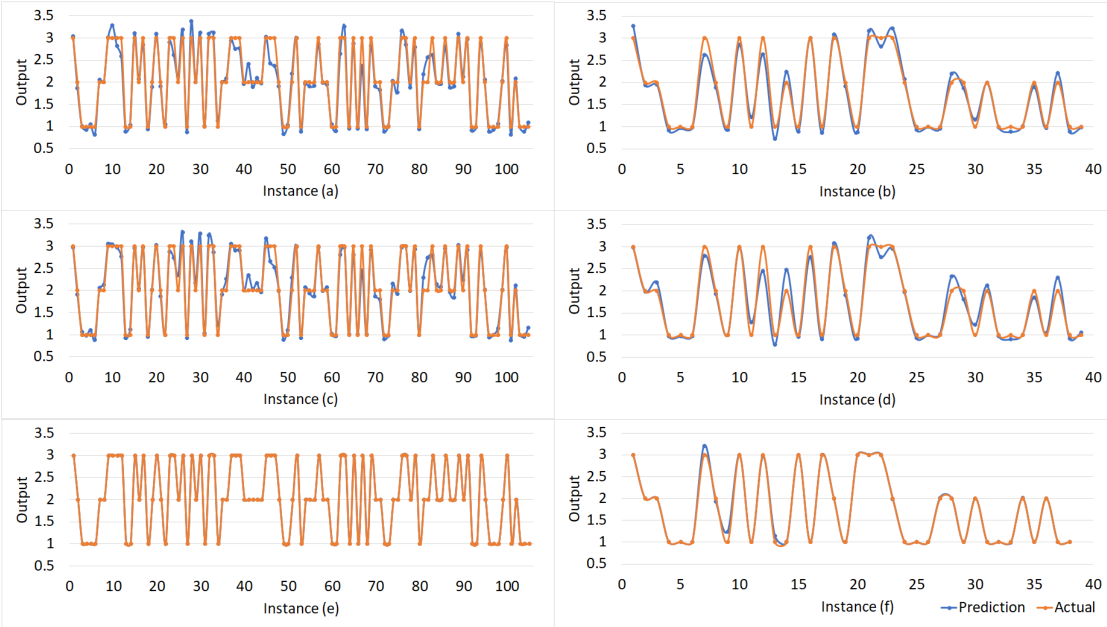


Figure 2.11: IRIS data set prediction vs actual outputs for (a) ANFIS-GA Training (b) ANFIS-GA Testing (c) ANFIS-PSO Training (d) ANFIS-PSO Testing (e) Cascaded ANFIS Training (f) Cascaded ANFIS Testing

number of example instances. Furthermore, these data sets are also different in the field of usage.

IRIS dataset is a combination of four feature inputs containing three classes. These classes are linearly separable. The breast cancer dataset contains nine attributes. These attributes are a mix of linear and nominal. The data of Statlog (Vehicle Silhouettes) is collected using elevated cameras. 18 features were obtained from the captured images and used as the dataset's attributes. The sport Activity Objective dataset uses 1000 sport-related articles with 59 attributes. The superconductivity dataset contains 81 attributes of superconductors and their relevant features. To predict whether a new molecule is a musk or non-musks, Musk 1 data set is introduced. This dataset is rich in 168 distinguished features. The human Activity Recognition Using Smartphones dataset includes 561 attributes, and these features were collected using a waist-mounted smartphone with inertial sensors.

The data sets mentioned above are used in the following manner. Each data set is divided into three parts. Such as, 60% of the data set instances are used for training purposes. The remaining instances are divided into two equal parts and used for testing and validating the algorithm.

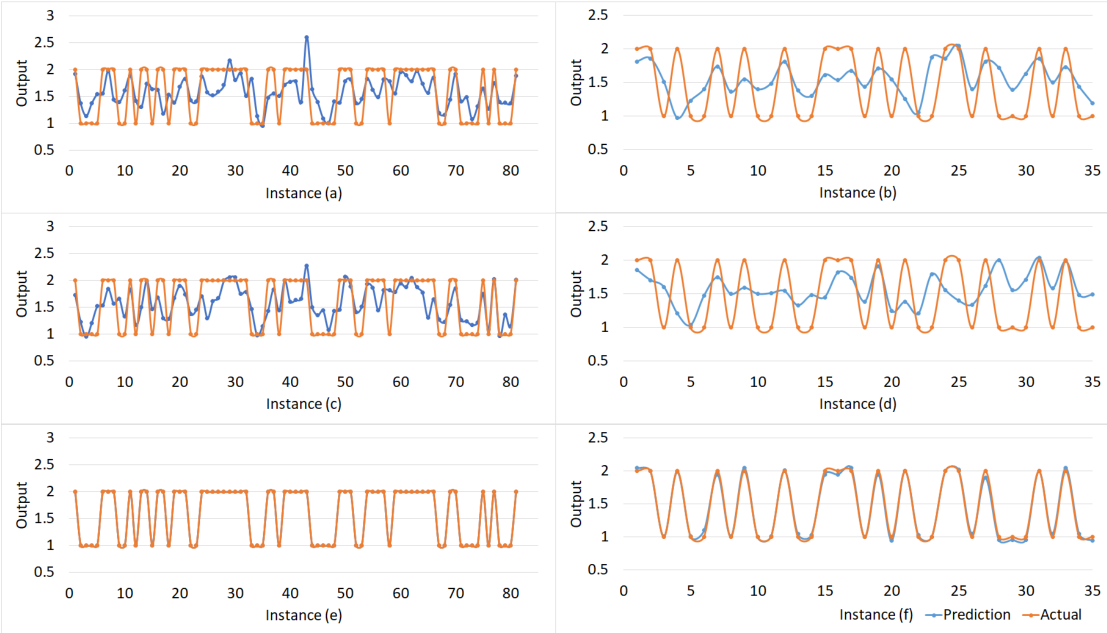


Figure 2.12: Breast Cancer data set prediction vs actual outputs for (a) ANFIS-GA Training (b) ANFIS-GA Testing (c) ANFIS-PSO Training (d) ANFIS-PSO Testing (e) Cascaded ANFIS Training (f) Cascaded ANFIS Testing

2.3.2 Comparative study against the Cascaded ANFIS algorithm

This research is conducted to present the effectiveness and accuracy of the novel Cascaded ANFIS model against three state-of-the-art algorithms:

1. Particle Swarm Optimization based Adaptive Neuro-Fuzzy Inference System (ANFIS-PSO)

2. Genetic Algorithm-based Adaptive Neuro-Fuzzy Inference System (ANFIS-GA)

As the record from the state-of-the-art methods, the above-mentioned hybrid methods outperform most of the other optimization algorithms [8–14]. Parameter adjustments are described in the next subsection.

2.3.3 Parameter Setting

As for the population, 30 is selected as in [15] for all the above mention algorithms. The number of iterations was set to 100, and the membership functions were limited to 4. PSO parameter setting is concluded as follows [16].

- Inertia Weight = 1

- Inertia weight damping ratio = 0.99

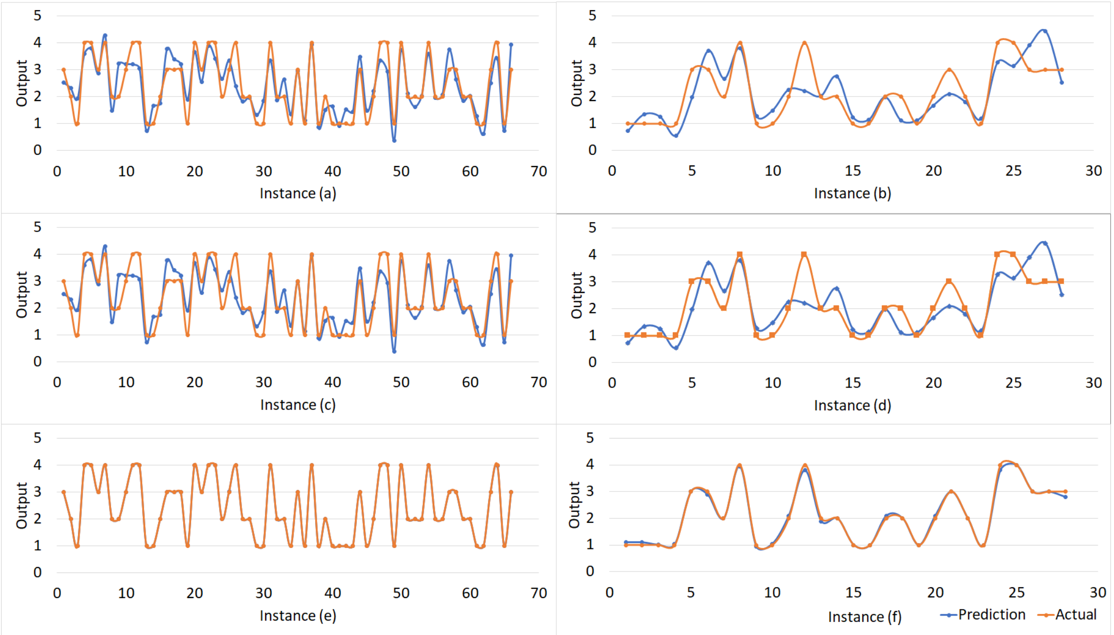


Figure 2.13: Vehicle data set prediction vs actual outputs for (a) ANFIS-GA Training (b) ANFIS-GA Testing (c) ANFIS-PSO Training (d) ANFIS-PSO Testing (e) Cascaded ANFIS Training (f) Cascaded ANFIS Testing

- Personal Learning Coefficient = 1
- Global Learning Coefficient = 2

Parameters of the GA are shown below [16].

- Crossover percentage = 0.7
- Mutation percentage = 0.5
- Mutation rate = 0.1
- Selection Pressure = 8
- Gamma = 0.2

2.4 Experimental Study and Results

In this section, a considerable amount of results are presented. The presented results can be divided into two main categories: the comparison against state-of-the-art algorithms and the iteration-wise comparison of the Cascaded ANFIS algorithm. As mentioned in the above section, ANFIS-PSO and ANFIS-GA were used for the comparison of the performance of a few data sets. The complexity of the data set is crucial in using state-of-the-art data sets because the size of the input dimensionality is proportional to the computational complexity.

2.4.1 ANFIS-PSO and ANFIS-GA vs Cascaded ANFIS

The experiments are carried out for a few data sets such as IRIS, Breast, and vehicle and the performance is calculated in the following ways.

$$MSE = \frac{1}{q}\sum_{t=1}^{q}(\bar{u}(t) - \hat{u}(t))^2 \tag{2.13}$$

$$RMSE = \sqrt{\frac{1}{q}\sum_{t=1}^{q}(\bar{u}(t) - \hat{u}(t))^2} \tag{2.14}$$

$$MAPE = \frac{1}{q}\sum_{t=1}^{q}\frac{\|\bar{u}(t) - \hat{u}(t)\|}{\|\hat{u}(t)\|} \times 100 \tag{2.15}$$

MSE is the mean squared error, and MAPE is the Mean Absolute Percentage Error. q is the size of the population, x_i is the error instance, and μ is the mean error. $\bar{u}(t)$ is the prediction and $\hat{u}(t)$ is the actual output. Moreover, the time consumption is recorded for each training and testing.

The first simulation was performed to present the accuracy of the novel algorithm. As mentioned above, three data sets were occupied for the comparison process. In Fig 2.6, RMSE is presented for the corresponding data sets and the algorithms. Here, It is clear that the novel algorithm outperforms the state-of-the-art algorithms in terms of accuracy.

Similarly, Fig 2.7,2.8, and 2.10 show the MSE, MAE, and MAPE, respectively. In each error measurement, the novel algorithm presents less error amount when compared to ANFIS-PSO and ANFIS-GA. It is worth mentioning that the novel algorithm has used only two iterations to reach these results. When advancing more iterations, better results can be observed. A detailed presentation is given in the next section.

Correlation is another measurement to recognize the performance of an algorithm. Figure 2.9 shows the correlation for three algorithms for three data sets. It can be observed that the novel algorithm obtained more significant results compared to ANFIS-PSO and ANFIS-GA. For the IRIS data set, ANFIS-PSO, ANFIS-GA, and Cascaded ANFIS are

0.977, 0.974, and 0.999, and Vehicle data set correlations are 0.865, 0.878, and 0.997, respectively.

The results mentioned above correspond to the training phase. Table 2.3 shows the errors and correlation of the IRIS data set at the testing phase. Results show that the RMSE of the novel algorithm for testing the IRIS data set is 0.005674, and the correlation is almost equal to 1. Tables 2.4 and 2.5 presents the testing results for Breast cancer and the vehicle data sets.

The prediction and actual outputs of the considered data sets are given in the following figures. Fig 2.11 shows the IRIS data set performance for the three algorithms. The smoothness of the outputs can be observed in the novel algorithm compared to the ANFIS-PSO and ANFIS-GA. The prediction curves against the actual output for breast cancer and vehicle data sets are given in Fig 2.12 and 2.13, respectively.

The time consumption of a system is a plain fact to present the computational complexity. Hence, the time consumption is obtained for each dataset training and testing. Tables 6 and 7 present the time consumption for testing and training, respectively.

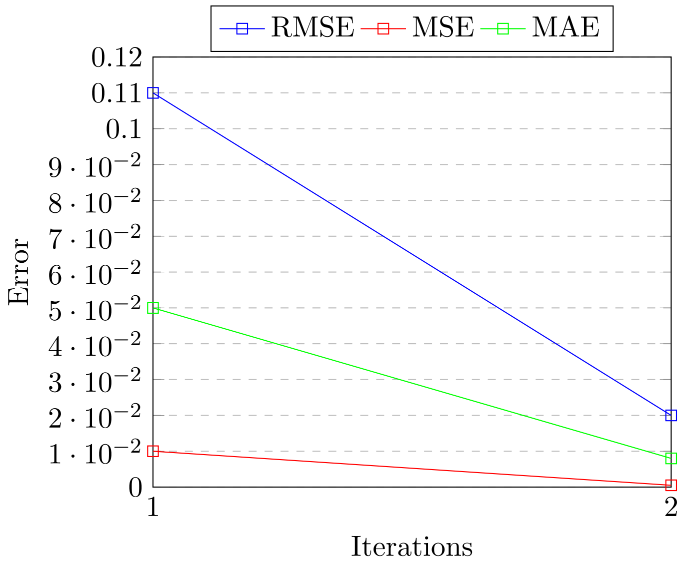


Figure 2.14: IRIS data set Error variance by the iterations

2.4.2 The novel Cascaded ANFIS performance evaluation

This section presents a detailed presentation of results for the Novel Cascaded ANFIS algorithm. As explained in the methodology section, the novel algorithm has iterations that can obtain far better results by increasing the number of iterations. Since this

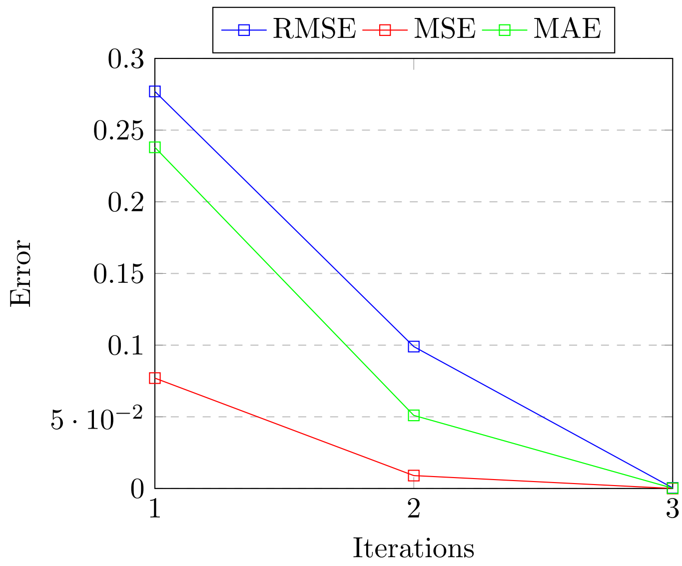


Figure 2.15: Breast Cancer data set Error variance by the iterations

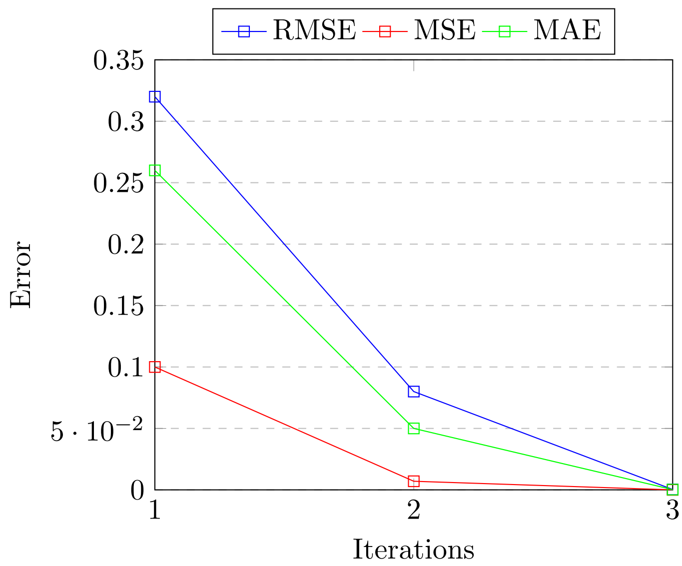


Figure 2.16: Vehicle data set Error variance by the iterations

study considers seven different data sets, the performance is presented in several graphs and plots.

Fig 2.14 shows the IRIS data set performance by iterations. As in the figure, in the first iteration, the RMSE of the novel algorithm is 0.114778533. Nevertheless, when it reaches iteration 2, the RMSE becomes 0.022531097. In Fig 2.15, the breast cancer data set is occupied, the number of iterations that have used is 3, and the RMSE decrements are 0.277640323, 0.099419942, and 0.000423648. Here, it can be observed that, at iteration 3, the RMSE is almost zero. Fig 2.16 presents the results for Vehicle data sets, and it has also used only three iterations.

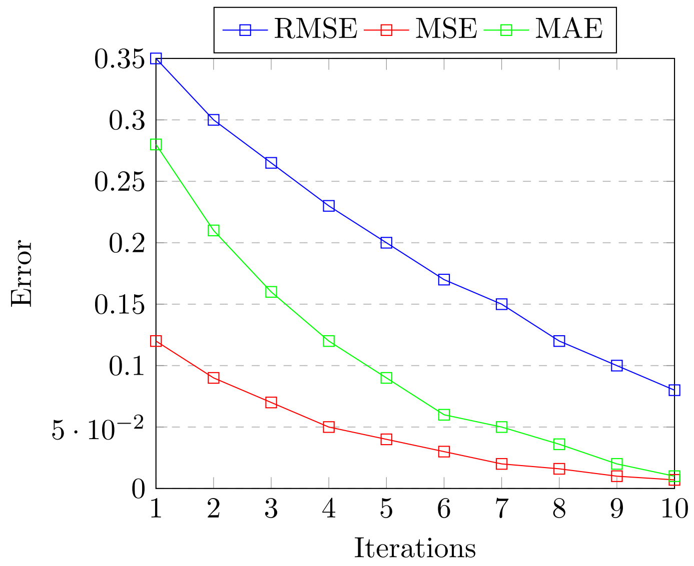


Figure 2.17: Sports articles for objectivity analysis data set Error variance by the iterations

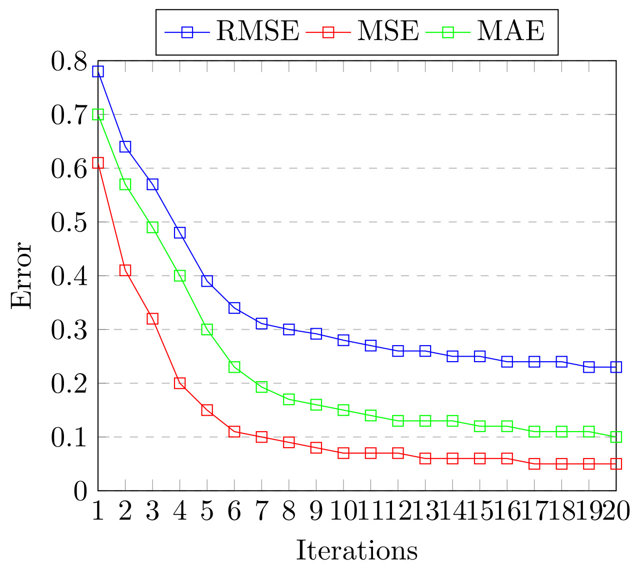


Figure 2.18: Human Activity Recognition Using Smartphones data set Error variance by the iterations

The performance results for Sports articles for objectivity analysis data set are shown in Fig 2.17. Here ten iterations are used in generating the minimum RMSE. Smart Phone Activities data set is presented in Fig 2.18. It has used 20 iterations and achieved 0.234818114 of RMSE.

Super Conductivity data set performance is shown in Fig 2.19. Though the number of input variables is higher than that of the Sport Article Objectives and Human Activity Recognition Using Smartphones, the iterations usage is surprisingly less for this data set. Within two iterations, the RMSE has reached 0.000396855. Musk 1 data set performance

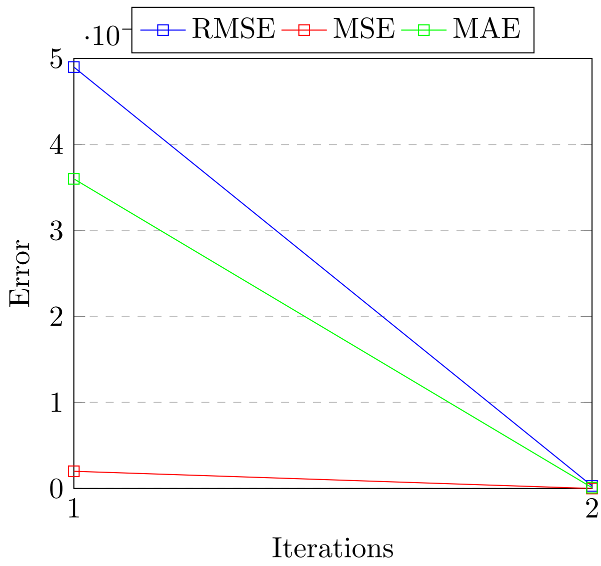


Figure 2.19: Super Conductivity data set Error variance by the iterations

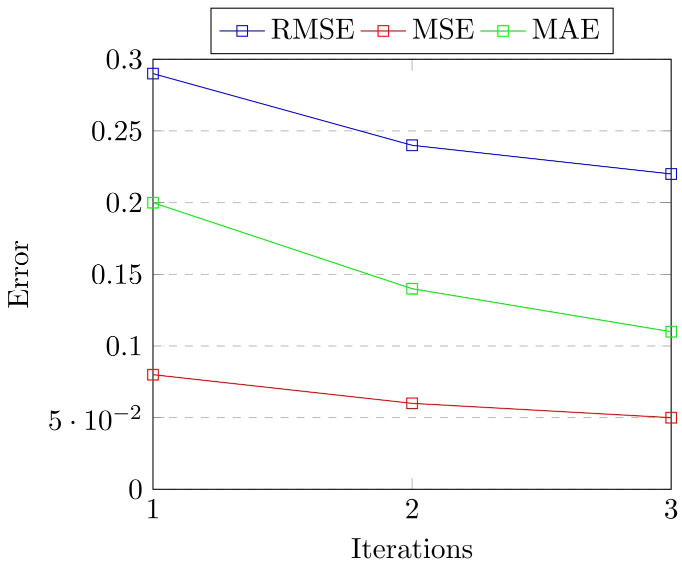


Figure 2.20: Musk 1 data set Error variance by the iterations

is presented in Fig 2.20. Here the error of 0.224424 is achieved within three iterations.

2.5 Discussion and Conclusion

This chapter presents the novel approach to the ANFIS algorithm. The main objective of this implementation is to solve the two main problems in the general ANFIS algorithm, namely the curse of dimensionality and the computational complexity. ANFIS is a well-known algorithm for the optimization of small input datasets. However, the novel algorithm is tested against two leading state-of-the-art algorithms with seven publicly available data sets. The input dimension of these datasets varies from 4 to 561. The main difficulty in state-of-the-art algorithms is using a more significant number of inputs. Increasing the number of inputs can significantly increase the computational complexity, and as a result, state-of-the-art algorithms can fail during optimization. However, the Cascaded ANFIS can use any number of inputs because, at any one time, it uses only two of the inputs. This is the main novelty of the Cascaded ANFIS algorithm. Moreover, the behaviour of the novel algorithm generates more accurate results with a lower error percentage.

Demonstration of the effectiveness of the Cascaded ANFIS algorithm was discussed using two parts. The first part is the effectiveness of the proposed algorithm against state-of-the-art algorithms, and the other part is the effectiveness of the proposed algorithms for a vast range of data sets. RMSE, MSE, MAE, MAPE, and Correlation are introduced to evaluate the results. In each aspect, the novel algorithm outperformed the state-of-the-art algorithms.

The input dimensions of the data sets vary from 4 to 561. As shown in figure 2.14,2.15, 2.16, 2.17, 2.18 and 2.19 the number of iteration usage changes regardless of the input dimension. For example, in figure 14, the IRIS data set has four input variables, and in the Cascaded ANFIS, the number of iteration usage is two. As well as the superconductivity data set iteration usage is two though it has 81 input variables. Therefore, the discrimination effectiveness of the variables affects the number of iterations. In figure 2.18, the error has been saturated around 0.25 after 12 iterations.

According to the experimental results, our Cascaded ANFIS algorithm shows significantly improved accuracy and can handle any number of inputs. A few significant limitations exist in implementing the Cascaded ANFIS algorithm on a microcontroller. Since the algorithm generates many variables, the microcontroller must have enough storage to store the data. Furthermore, it is not possible to use the online training method using the novel algorithm because the time consumption is considerably higher than the online trainable algorithms.

The following chapters will discuss the Cascaded-ANFIS-based applications and other investigations carried out during the doctoral study.

References

[1] Namal Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. A novel optimization algorithm: Cascaded adaptive neuro-fuzzy inference system. *International Journal*

of Fuzzy Systems, 23(7):1955–1971, 2021.

[2] Alireza Baghban. Application of the anfis strategy to estimate vaporization enthalpies of petroleum fractions and pure hydrocarbons. *Petroleum Science and Technology*, 34(15):1359–1366, 2016.

[3] Lei Ye and Xiangjun Ou. Spatial-temporal analysis of daily air quality index in the yangtze river delta region of china during 2014 and 2016. *Chinese Geographical Science*, 29(3):382–393, 2019.

[4] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1):116–132, 1985.

[5] Shahaboddin Shamshirband, Masoud Hadipoor, Alireza Baghban, Amir Mosavi, Jozsef Bukor, and Annamária R Várkonyi-Kóczy. Developing an anfis-pso model to predict mercury emissions in combustion flue gases. *Mathematics*, 7(10):965, 2019.

[6] Mohd Najib Mohd Salleh, Noureen Talpur, and Kashif Hussain. Adaptive neuro-fuzzy inference system: Overview, strengths, limitations, and solutions. In *International Conference on Data Mining and Big Data*, pages 527–535. Springer, 2017.

[7] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[8] Afshin Tatar, Ali Barati-Harooni, Adel Najafi-Marghmaleki, Behzad Norouzi-Farimani, and Amir H Mohammadi. Predictive model based on anfis for estimation of thermal conductivity of carbon dioxide. *Journal of Molecular Liquids*, 224:1266–1274, 2016.

[9] Osman Taylan and Bahattin Karagözoğlu. An adaptive neuro-fuzzy model for prediction of student's academic performance. *Computers & Industrial Engineering*, 57(3):732–741, 2009.

[10] Piotr Dziwiński and Łukasz Bartczuk. A new hybrid particle swarm optimization and genetic algorithm method controlled by fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 2019.

[11] Harihar Kalia, Satchidananda Dehuri, Ashish Ghosh, and Sung-Bae Cho. Surrogate-assisted multi-objective genetic algorithms for fuzzy rule-based classification. *International Journal of Fuzzy Systems*, 20(6):1938–1955, 2018.

[12] Tien-Loc Le, Tuan-Tu Huynh, and Chih-Min Lin. Self-evolving interval type-2 wavelet cerebellar model articulation control design for uncertain nonlinear systems using pso. *International Journal of Fuzzy Systems*, 21(8):2524–2541, 2019.

[13] MH Fazel Zarandi and F Dorry. A hybrid fuzzy pso algorithm for solving steelmaking-continuous casting scheduling problem. *International Journal of Fuzzy Systems*, 20(1):219–235, 2018.

[14] Chih-Min Lin and Tien-Loc Le. Pso-self-organizing interval type-2 fuzzy neural network for antilock braking systems. *International Journal of Fuzzy Systems*, 19 (5):1362–1374, 2017.

[15] Talha Ali Khan, Khawaja Zain-Ul-Abideen, and Sai Ho Ling. A modified particle swarm optimization algorithm used for feature selection of uci biomedical data sets. In *60th International Scientific Conference on Information Technology and Management Science of Riga Technical University, ITMS 2019-Proceedings*, 2019.

[16] Ibrahim M Alarifi, Hoang M Nguyen, Ali Naderi Bakhtiyari, and Amin Asadi. Feasibility of anfis-pso and anfis-ga models in predicting thermophysical properties of al2o3-mwcnt/oil hybrid nanofluid. *Materials*, 12(21):3628, 2019.

Chapter 3

Classification of the Fruit-360 dataset

Chapter 3 showcases an application developed using the proposed algorithm - Cascaded-ANFIS. In this application, a new structure is introduced for situations where the number of input features is enormous. This work is published in Sensors (MDPI) [1].

3.1 Introduction

Given the tremendous growth of the current population rate, the foods we consume are a significant concern. Fruits are an essential consuming food in the day-to-day life pattern of most people and a highly recommended source of nutrient supply by nutritionists. Various strategies for fruit detection utilizing computer vision technologies have been used for many years. These approaches are used to categorize and distinguish various types of fruits from a collection of photographs. Fruit categorization is still a contentious and complex problem in the research and practising industries. For example, identifying the class of a particular fruit allows grocery staff to calculate its price [2] quickly. Furthermore, nutritional recommendations are beneficial in assisting consumers in picking appropriate food varieties that satisfy their nutrient and well-being demands [3, 4]. Fruit categorization techniques are frequently employed in most food facilities for automated packing.

The fruit types and sub-types are location-dependent (vary from location to location, even in the same country). Thus, manual fruit categorization is still a challenging problem. This vast disparity is centred on the availability of population-dependent, and region-dependent fruits and the required elements in the fruits [4]. Artificial Intelligence (AI) and Machine Learning (ML) approaches are utilized in various applications to give optimal solutions to challenges faced in a variety of disciplines such as image analysis, speech recognition, forecasting, prediction, massive dataset analysis, and marketing [5]. Thus, the rapid advancement in computer vision and machine learning, particularly in the recent decade, has drawn the attention of various researchers to the use of established approaches in automatic fruit categorization. Researchers frequently employed elements linked to exterior quality descriptors in their study, such as form, size, texture, and colour [6, 7]. Most of the suggested classifiers were either constrained to a certain kind of fruit or showcased poor accuracy. Many of the classification systems are purely based on Neural Network (NN) algorithms, and very few approaches are in the literature based on Fuzzy Logic (FL).

3.2 Related Works

Experts introduced several automatic fruit and vegetable categorization algorithms in recent years. The VeggieVision [8] was the first product from a significant attempt to recognize vegetables and fruits. This device had an integrated scale as well as a digital camera. The camera captures the image when an item is placed on the scale. Color, texture, and other characteristics were retrieved and compared to previously stored characteristics of distinct product varieties. These stored characteristics were acquired throughout the training procedure. When the training and testing datasets were from the same store, the best pick had a classification accuracy of 82.6%. The classification accuracy decreased dramatically when the training and testing datasets were from different stores.

Seng and Mirisaee [9] suggested another fruit detection method based on colour, shape, and size. The colour was represented by the mean RGB value, shape by the measure of roundness, and size by the area and perimeter values. These feature values were then used to classify data using the k-nearest neighbour technique. Despite the excellent accuracy rates reported, the training and testing datasets were relatively small.

Wang et al. [10] proposed two different machine learning-based fruit categorisation algorithms. Wavelet entropy, principal component analysis, feed-forward neural networks trained with Fitness-Scaled Chaotic Artificial Bee Colony (FSCABC), and biogeography-based optimization techniques were used in their procedures. The categorization accuracy for both approaches was 89.5% which is higher than the earlier approaches. However, Pennington and Fisher [4] were the first scientists to utilize the clustering approach to categorize fruits and vegetables in 2009. They have employed a dataset having 104 common fruits and vegetables for classification. Visible spectroscopy was used by Pholpho et al. [11] to distinguish damaged and undamaged fruits. Furthermore, Yang et al. [12] presented an estimating approach for blueberry fruit identification using multispectral image analysis. In contrast, computer vision and multi-class Support Vector Machine (SVM) were used to categorize distinct varieties of fruit with an 88.20% accuracy [13]. Later, eight citrus fruits were identified using Raman spectroscopy as a quick and non-destructive measure using two analytic approaches (hierarchical cluster and principal component) [6]. In addition, Fadhel et al. [14] employed colour segmentation to identify immature strawberries. They have used two different methods for classification: colour thresholding and K-means clustering. The results indicate that the colour thresholding results outperform the clustering method. Literature also presents the related studies in impurity identification in olive oil using similar techniques of computer vision and machine learning [7]. Furthermore, Breijo et al. [15] used an electronic nose (also known as a piece of olfactory sampling equipment) to characterize the odour of Diospyros kaki (Persimmons). The system's operating parameters can impact the changeable configurations, allowing the system to be flexible.

On the other hand, Fan et al. [16] used an artificial neural network with two hidden layers to predict the texture features derived from a food-surface picture. The back-propagation method was utilized for training the neural network. However, the neural network approach had some disadvantages, including behaving as a black box, intensive duration of development, and the requirement of a lot of data.

However, Omid et al. [17] presented an expert system for extracting size and defect information using machine vision and fuzzy logic. This approach employed two types of membership functions, including triangular and trapezoidal. In addition, the study was evaluated using the correct classification rate (CCR), and overall accuracy of 95.40% was obtained.

Another automatic fruit categorization system was proposed based on the fitness-scaled chaotic artificial bee colony algorithm [18]. The authors have compared the performance of the proposed algorithm with three well-known AI methods. However, the proposed FSCABC-FNN method has only shown an 89.10% accuracy outperforming other algorithms. In addition, Khanmohammadi et al. [19] have proposed a classification method based on near-infrared spectrometry (FT-NIR) and Square SVM. They have succeeded in obtaining a miner prediction error rate of 2%.

Furthermore, a texture-based method that involves descriptor computation and interest-point feature extraction was proposed [20]. They stated that the study shows excellent results on a single image detection rate having 85.00% and 100.00% for pineapple and bitter lemon fruits, respectively. Date fruits were identified using Weber's local descriptor and local binary pattern approaches and SVM for classifier and Fisher discrimination ratio for feature selection [21]. This study has considered three feature descriptors: colour, texture, and shape. The proposed algorithm shows a 98.00% accuracy after the dimension reduction using Fisher Discrimination Ratio (FDR).

The literature presents many related research studies based on convolutional neural networks (CNN) to the Fruit-360 dataset in recognizing the fruits. A CNN-based VGG16 model used developed by Siddiqi [22] to classify 72 classes of the Fruit-360 dataset, and the author has obtained 99.27% accuracy in total. Ghazanfar et al. [23] have presented a model using deep convolutional neural networks (DCNN) to classify the same dataset (Fruit-360) and acquired a 92.00% recognition rate. The individual classes of the Fruit-360 dataset were combined to create new classes in this research. Therefore, the total number of categories was reduced to 16 for the classification. This created the problem robustly. In addition, Ghosh et al. [24] have introduced an image classification model using ShufleNet V2 that is based on the CNN algorithm. They have obtained an accuracy of 96.24% for 40 classes in the Fruit-360 dataset. Furthermore, Postalcıoğlu [25] has also presented a model based on CNN. Three different optimizers, including, Stochastic Gradient Descent with Momentum (SDGM), Adaptive Moment Estimation (Adam), and Root Mean Square Propagation (RMSPROP), were used in that analysis to evaluate the results. The results were 98.08%, 98.83%, and 99.02%, accurate respectively. However, the research was only conducted for 48 classes in the Fruit-360 dataset. Therefore, the study was not a comprehensive work. Another study by Ziliang et al. [26] have showcased an accuracy of 98.06% for a classification model using the CNN algorithm. However, they have extended the analysis for 81 classes of the Fruit-360 dataset.

The in-depth review of related literature presents the following drawbacks and shortcomings.

1. The studies required expensive sensors such as weight, dew, heat, chemical, gas-sensitive, and infrared light to model the classification.

2. The classifiers are only capable of recognizing a few types of fruits, not the whole Fruit-360 dataset.

3. The system performance is insufficient, owing primarily to closely related texture, colour, and shape properties.

4. The classification precision falls short of the standards for typical applications.

5. The algorithms required a higher computational power.

Therefore, this research study proposes a new algorithm based on the Cascaded Adaptive Neuro-Fuzzy Inference System (Cascaded-ANFIS) [27] to fill the above-identified research gaps in the literature and then to present a much enhanced and robust model to identify the fruits based on their properties. The significant contributions of the presented research can be listed as follows.

1. This study proposes a novel structure for the Cascaded-ANFIS algorithm for Image Classification.

2. The system is designed using nine state-of-the-art feature descriptors (including Colour Structure (CS), Region Shape (RS), Edge Histogram (EH), Column Layout (CL), Gray-Level Co-Occurrence Matrix (GLCM), Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Histogram of Oriented Gradients (HOG), and Oriented FAST and rotated BRIEF features (ORB)).

3. The total dataset of 131 classes is used for the classification.

4. The novel system can reduce the dimension input to different features due to the usage of the feature reduction method.

5. Comparison of the accuracy with the state-of-the-art algorithms (including CNN with Stochastic gradient descent with momentum, CNN with adaptive moment estimation, CNN with RMS propagation, Customized Inception V3, Customized VGG 16, Customized MobileNet, Vanilla MobileNet, ShufeNet V2, DCNN, and ResNet18).

6. The comparative computational power is relatively inexpensive while providing an accuracy up to 98.36%.

3.3 Proposed Methodology

3.3.1 The Fruit-360 Dataset

Fruits-360 is a dataset with 90,483 fruit photos (67692 in the training set and 22688 in the test set) [28]. The collection contains 131 different varieties of fruits, and each has an image capturing only one fruit. These images are 100×100 pixels in size. Each fruit type's training and test sets contain a somewhat different number of photos. However, in most situations, roughly 70% training and 30% test images are provided for each fruit type. These images are obtained by filming a brief fruit video for twenty seconds while it is slowly spun by a motor and extracting frames/images from that movie. A white sheet of paper is used as the background for the capture. A specific algorithm then eliminates the background of each fruit. The varying light intensity can impact the background; therefore, it has to be removed.

This study is based on image data classification, which provides many input dimensions to the system. Therefore, a state-of-the-art dimension reduction method was investigated to solve this issue. Hence, three well-known dimension reduction (DR) methods were considered: Independent Component Analysis (ICA) [29], Principle Component Analysis (PCA) [30, 31], and Multi-Dimensional Scaling (MDS) [32].

The results of using these DR methods are illustrated in the Implementation of the algorithm section. A simple experiment was conducted to identify the best algorithm for dimension reduction. Three well-known datasets (vehicles by Siebert [33], breast cancers by Wolberg and Mangasarian [34], Musk 1 by Dietterich et al. [35]) were used to reduce the dimension using the Cascaded-ANFIS algorithm using all three methods. These datasets were selected based on different perspectives, such as the field of interest and the number of inputs and outputs.

3.3.2 Image Data Analysis – Feature Extraction

Features are the key ingredient in implementing a classifier. Therefore, according to the literature, nine feature descriptors are used to extract different features from the Fruit-360 image dataset. This section provides a brief introduction to each of these feature extraction methods. The first method is the Color Structure descriptor. It is based on histogram equalization, but it seeks and gives a complete description by differentiating localized colour variations for each colour [36]. The next feature descriptor is the Region Shape. The shape characteristics are less developed than their colour and texture equivalents because of the intrinsic difficulties of portraying forms [37].

However, due to the variety of possible projections of a 3D object into 2D shapes, the complexity of each object shape, the presence of shadows, occlusions, non-uniform illumination, and uneven surface reflectivity, it is not accessible to precisely segment an image into meaningful regions using low-level features. Therefore, the Column Layout feature descriptor was used as the third feature extraction method.

Edge Histogram descriptor (EDH) represents the geometry of an image and is meant to depict the distribution of local edges inside pictures [38]. Therefore, this research used the EDH descriptor as the fourth feature extraction method. Edges are essential for viewing image information, and the histogram was used to characterize them. The homogeneous colour histogram and texture feature cannot reproduce an image's EDH-described qualities [39, 40]. The fifth feature descriptor is the Gray Level Co-occurrence Matrix (GLCM). It determines how frequently unique combinations of grey levels co-occur in an image or section of an image given an image made up of pixels, each with an intensity (a specific grey level). The GLCM contents are utilized in texture feature calculations to measure the change in intensity (also known as image texture) at the pixel of interest [41].

The sixth and seventh descriptors are Scale Invariant Feature Transform (SIFT) [42] and Speeded up Robust Features (SuRF) [43]. SIFT characteristics include scale and rotation invariance, and they have various advantages, including localization, distinctiveness, quantity, efficiency, and flexibility. On the other hand, SURF is a quick and trustworthy approach for encoding and estimating pictures in a local, similarity-invariant way. The SuRF technique's main appeal is its ability to calculate operators fast using box filters, enabling real-time tracking and object recognition applications.

The Histogram of Oriented Gradients (HOG) feature descriptor is the eighth feature extraction method used in this study. It is related to the Canny Edge Detector and the SIFT, and it is used in image processing to detect objects [44]. The method counts how often a gradient orientation appears in a specific picture section. The ninth and the last feature descriptor was presented by Ethan et al., and it is called the Oriented FAST and Rotated BRIEF (ORB) [45]. The FAST key-point detector serves as the foundation for the ORB descriptor. ORB performs feature identification similarly to SIFT and SURF while being roughly two orders of magnitude faster. Because of its significant contributions, the ORB descriptor is employed as the feature extractor in many machine learning models [46, 47].

3.3.3 Application Methodology – Novel Modified Structure for the Cascaded-ANFIS

The flowchart for the developed Cascased-ANFIS algorithm is presented in Figure 3.1. In this figure, $A_{(i,j)}$ represents the ANFIS structure, and i and j are the number of the levels and the number of the ANFIS structures in the corresponding level. Hence, seven ANFIS structures in the first level are represented as $A_{1,1}...A_{1,7}$.

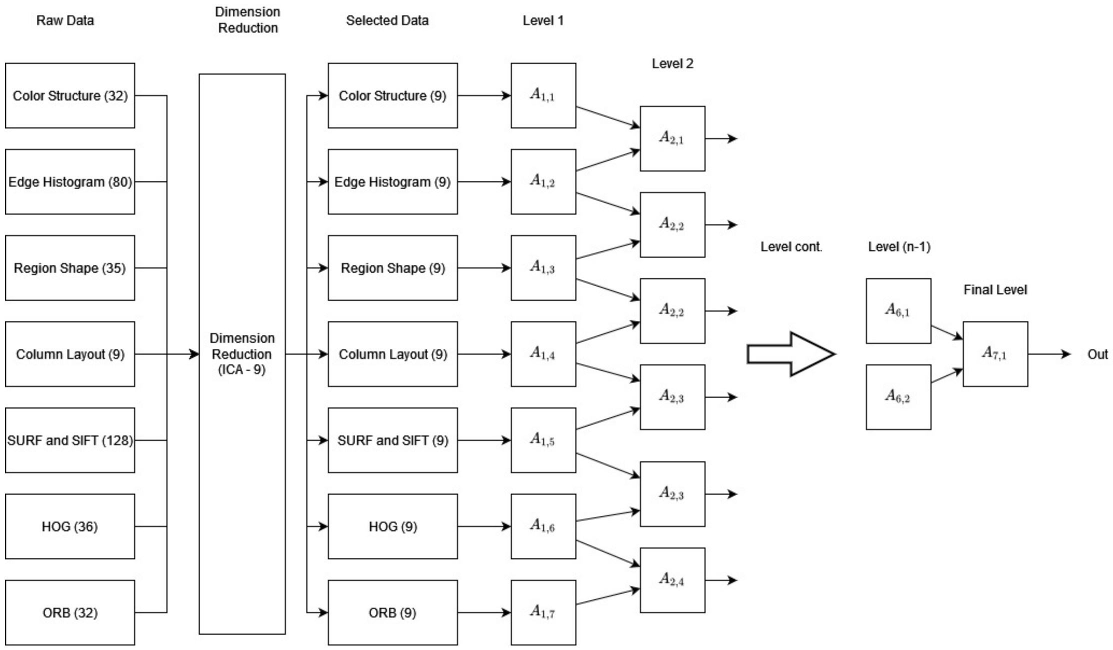


Figure 3.1: The Proposed Modified Novel Structure of Cascaded-ANFIS algorithm.

A modified Cascaded-ANFIS algorithm had to be built to extract more features from a few descriptors from a single image (for example, 352 features can be extracted, as shown in Figure 3.1). However, reducing these features to fewer meaningful features is challenging.

One of the main aspirations of this study is to generate a real-time system with higher accuracy compared to the existing algorithms. Therefore, reducing as many input dimensions as possible gives an added advantage in reducing time consumption and computational complexity. The reduction of input features was carried out considering each feature descriptor individually.

As shown in Figure 3.1, the developed model used ICA as the feature reduction method. The feature reduction was carried out for each set individually. The resulting feature set is the 'Selected Data' (refer to Figure 3.1), and each set of features contains nine features. Using ICA, the feature number was reduced to 63 (from 352). Then, the initial level of the modified Cascaded-ANFIS algorithm was started. The initial level of this structure uses seven inputs (consisting of 9 features) even though the usual Cascaded-ANFIS algorithm uses two inputs. Therefore, this modified architecture has seven Cascaded-ANFIS levels. Each ANFIS uses the previous output as the input to the current module.

3.3.4 Performance Analysis Techniques

The performance of the developed model was analysed using a confusion matrix. A confusion matrix gives information about the predictions. Other classification matrices shown in Equations 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, and 3.7 were tested to understand the confusion matrix.

$$Accuracy_{Avg} = \frac{\sum_{i=1}^{l} \frac{tp_i + tp_i}{tp_i + fn_i + fp_i + tn_i}}{l} \tag{3.1}$$

$$Precision_{\mu} = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fp_i)} \tag{3.2}$$

$$Recall_{\mu} = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fn_i)} \tag{3.3}$$

$$FScore_{\mu} = \frac{(\beta^2 + 1)Precision_{\mu}Recall_{\mu}}{\beta^2 Precision_{\mu} + Recall_{\mu}} \tag{3.4}$$

$$Precision_{M} = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l} \tag{3.5}$$

$$Recall_{M} = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l} \tag{3.6}$$

$$FScore_M = \frac{(\beta^2 + 1)Precision_M Recall_M}{\beta^2 Precision_M + Recall_M} \tag{3.7}$$

tp_i, tn_i, fp_i, and fn_i are truly positive, true negative, false positive, and false negative, respectively. In addition, l is the total number of classes, and μ and M are the micro and micro-averaging. Each of these parameters conveys valuable information about the performance of the classification when the problem is multi-class [48]. The performance of the novel Cascaded-ANFIS model was tested and presented for its accuracy.

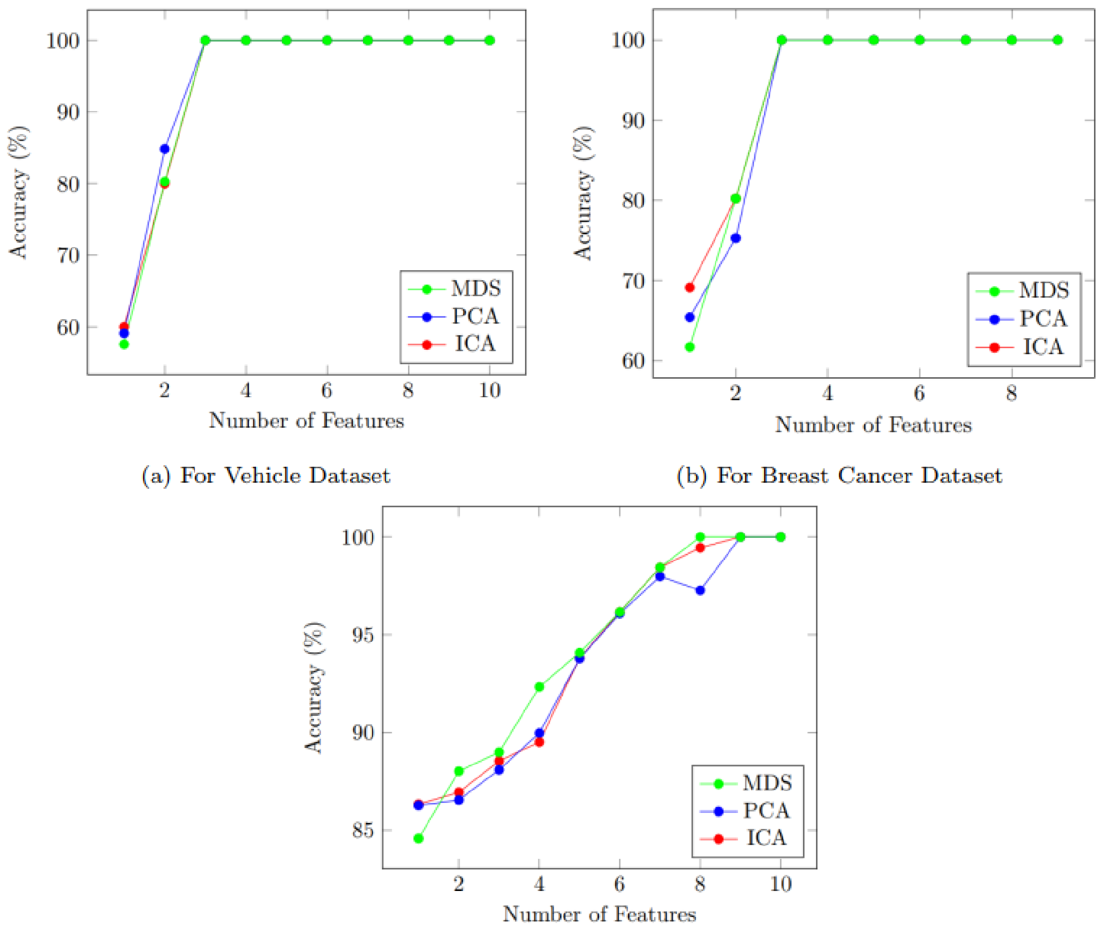
3.4 Results and Discussion

3.4.1 Feature dimension reduction

Figures 3a, b, and c show the dimension reduction results for three algorithms. They all reach very good accuracies and showcase similar variations. Therefore, the time consumption to perform these algorithms were considered the selection criterion.
Figure 4 presents the time consumption to perform the DRs. As can be seen, the time consumption is almost similar in ICA and PCA from features 2 to 10. However, MDS shows a longer calculation time when compared with the other two methods. For example, 0.99 s, 1.07 s, and 192.76 s were consumed at feature number two for ICA, PCA and MDS, respectively.
However, with the increase in feature numbers, ICA and PCA have caught up with MDS. This can be seen in feature number 10. Nevertheless, ICA still shows better performance with respect to time consumption. At feature number 10, the time consumption is recorded as 1632 s, 1710 s, and 1852 s for ICA, PCA, and MDS. Therefore, the modified Cascaded-ANFIS algorithm was constructed using the ICA feature dimension reduction method.

3.4.2 Learning behaviour by iterations

The learning behaviour performance of selected algorithms was compared with the novel Cascaded-ANFIS algorithm performances. A summary of the experiment is shown in Table 3.1.



(a) For Vehicle Dataset

(b) For Breast Cancer Dataset

(c) For Musk 1 Dataset

Figure 3.2: Accuracy comparison of feature dimension reduction algorithms when used on well-known datasets (Breast Cancer, Vehicle, and Musk 1).

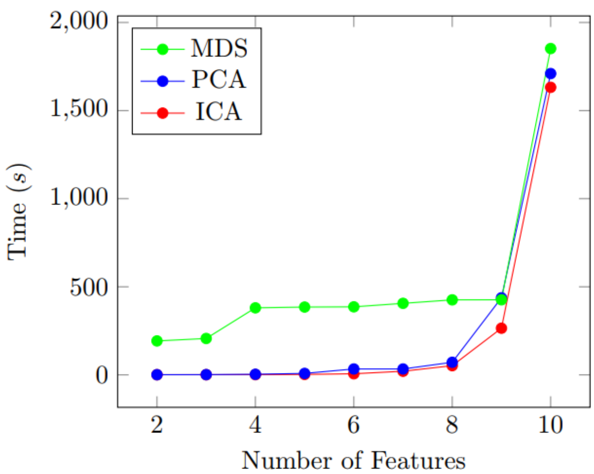


Figure 3.3: Time consumption for feature dimension reduction (Time is denoted in seconds (s)).

Table 3.1: Model training performance with iterations.

No of Iterations	SVM	MLP	ANFIS	PSO-ANFIS	GA-ANFIS	Cascaded-ANFIS
1	1.98	3.28	2.02	1.91	1.92	0.31
10	1.61	0.95	2.02	1.91	1.92	0.24
100	1.20	0.43	2.02	1.43	1.83	0.20

Table 3.2: Sample Distribution of Fruit-360 dataset among some of the classes

Class ID	Class Label	Number of Samples
0	Apple Braeburn	492
12	Apple Red Yellow - 2	672
25	Cauliflower	702
32	Chestnut	450
42	Ginger Root	297
44	Grape Blue	984
66	Mangostan	300
73	Nut Pecan	534

The traditional non-fuzzy-based algorithms, such as SVM and MLP, showed a decreasing trend in RMSE, while the fuzzy-based algorithms show a neutral behaviour to increasing iterations. In addition, GA-ANFIS and PSO-ANFIS algorithms presented a decreasing trend of RMSE after many iterations. However, it is noteworthy that the ANFIS algorithm kept the RMSE at 2.02 during all iterations, while the Cascaded-ANFIS algorithm gives the best RMSE. The Cascaded-ANFIS algorithm trains several FIS modules at a single iteration. Therefore, it is clear that the Cascaded ANFIS reaches a lower RMSE value in fewer iterations. Hence, this proves that the Cascaded-ANFIS algorithm performance saturates at the minimum number of iterations.

3.4.3 Confusion matrix analysis

The performance comparison of the modified Cascaded-ANFIS structure was evaluated using learning curves and the confusion matrix analysis. The overall confusion matrix was generated for all 131 classes when using the Cascaded-ANFIS algorithm for the classification. The resulting confusion matrix is given in Figure 6.

Due to the unbalance samples in each class of the Fruit-360 dataset, the confusion matrix shows different colours at the top predictions. Therefore, summarized class information is given in Table 1 for further clarification.

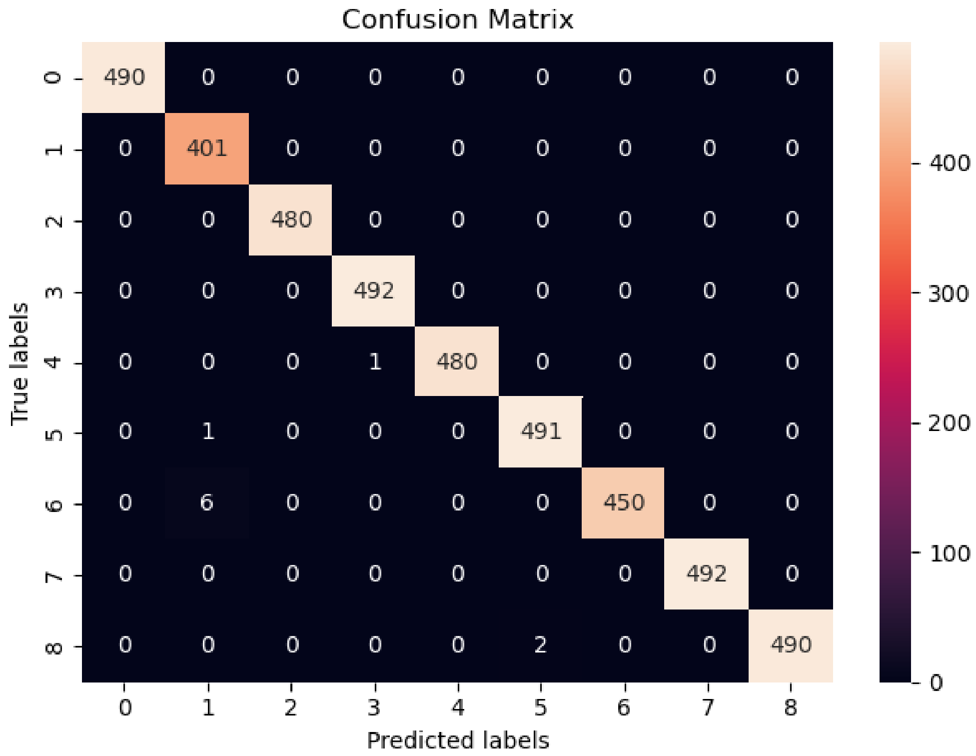


Figure 3.4: Confusion Matrix for eight classes classification.

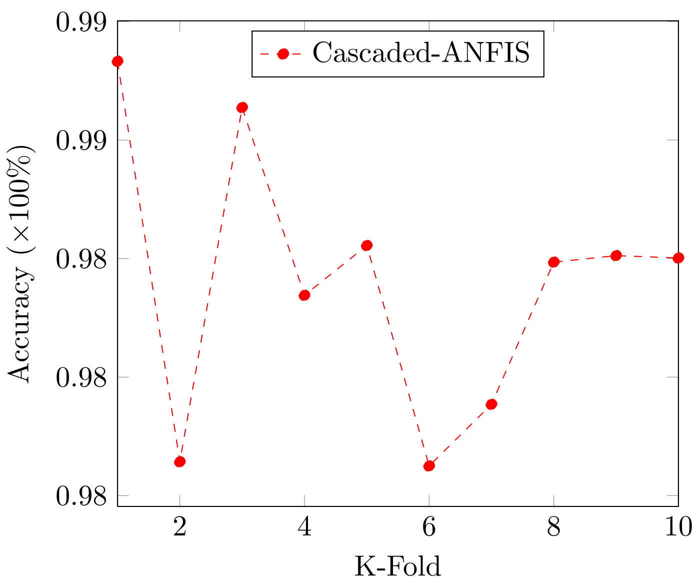


Figure 3.5: 10-Fold Cross-Validation of the Accuracy of the Classifications

Moreover, 10-fold cross-validation was carried out to investigate the stability and robustness of the proposed algorithm. Figure 10.2 shows the resulting plot of 10-fold cross-validation. As shown in the figure, the accuracy remains between 98% and 99%. However, the average accuracy is calculated as 98.36%.

Table 3.3: Performance of Confusion Parameters

Metric	Performance Value
Average Accuracy	0.9841
Precision$_{\mu}$	0.9841
Recall$_{\mu}$	0.9841
FScore$_{\mu}$	0.9841
Precision$_{M}$	0.9846
Recall$_{M}$	0.9849
FScore$_{M}$	0.9845
Precision$_{W}$	0.9843
Recall$_{W}$	0.9841
FScore$_{W}$	0.9840

3.4.3.1 The accuracy evaluation of the confusion matrix

The class prediction accuracy was tested as a percentage of correctly predicted vs total tested images. A classification accuracy of 98.41% was achieved from the developed Cascaded-ANFIS model. In addition, the accuracy was checked for the confusion matrix using Equations 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, and 3.7. The dataset class samples were not balanced in the Fruit-360. Thus, the confusion matrix was generated for all 131 classes. Figure 3.4 presents a sample confusion matrix of eight classes. Table 2 shows the performance values for each parameter of the Cascaded-ANFIS algorithm-based classifier.

Four main parameters can be extracted from a confusion matrix as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP is the value of correct predictions of positives out of actual positive samples, whereas FP is the false positive representations of actual negative samples. TN is the accurate pessimistic prediction of actual negative samples, and FN is the false-negative samples. When the classes are unbalanced, the recall score is a good indicator of prediction success. It is the proportion of TP to a genuinely positive FN in mathematics.

As can be seen in Table 3.3, all the parameters are above the level of 0.98. This concludes that the classification performance of the Cascaded-ANFIS model is excellent and served well for the Fruit-360 dataset.

Table 3.4: Configuration of the host computer.

Processor	Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz 3.70 GHz
Installed RAM	64.0 GB (63.9 GB usable)
Windows Edition	Windows 10 Education
HDD	4TB
SSD	1TB

Table 3.5: Comparison of classification accuracy against similar research work

Reference Study	Algorithm	Size of the Dataset		Test Accuracy
		# classes	# samples	
Seda Postalcioglu (2019)[25]	CNN withStochastic gradient descent withmonentum	48	50590	98.08
	CNN with Adaptive moment estimation			98.83
	CNN with Root Mean Square Propagation			99.02
Raheel Siddiqi (2019)[22]	Customized Inception v3	72	48249	99.1
	Customized VGG16			99.27
Ziliang Huang et al. (2019)[26]	Customized MobileNet	81	55244	98.06
	Vanilla MobileNet			95.98
Sourodip Ghosh et al. (2020)[24]	ShufeNet V2	31	29347	96.24
Ghazanfar Latif et al. (2020)[23]	DCNN	18	22341	95
Jorg Martinet al. (2019)[49]	ResNet18	116	58428	98.7
This Study(2022)	Cascaded-ANFIS	131	67692	98.36

3.4.4 Comparison of classification accuracy against state-of-the-art algorithms

Literature showcases several attempts at classifying the Fruit-360 dataset at different years. The dataset is upgraded year by year. Thus the usage of classes differs from study to study. Table 3.5 shows the best attempts in the past using different algorithms.

Ten different algorithms used to classify Fruit-360 data into its classes are summarized in Table 4. It is worth noting that these algorithms are based on CNN, such as CNN with stochastic gradient descent with momentum, CNN with adaptive moment estimation, and Customized Inception V3. Importantly, all these attempts have been made during 2019 and 2020.

The accuracy was measured as a percentage for all cases. The best results were found for the CNN approach when employing a Customized VGG16 network [22], which was 99.27%. Though the results are higher in that research than in the proposed method in this study, the amount of data in the Fruit-360 dataset was lesser for Siddiqi [22]. Only 72 classes with 48,249 samples were used by Siddiqi, while 131 types with 90,380 pieces were utilized for the presented study.

In addition, the attempts presented in the year 2020 have a noticeable accuracy reduction for 4-5% (96.25% [23] and 95% [24]). This could be due to the growth of the dataset. Therefore, the Cascaded-ANFIS model with a larger dataset has advantages in classification accuracy.

Usually, an expensive GPU is needed to run CNN-based algorithms due to high computational cost, whereas a conventional computer without GPU is enough for fuzzy-based ANFIS algorithms. Our experiments showed that the Cascaded-ANFIS algorithm could be implemented successfully using a computer without GPU, as shown in Table 3.

Moreover, CNN-based methods use an inbuilt feature extraction method, and the classification processes are performed using a fully connected neural network. However, the Cascaded-ANFIS study uses a fuzzy-based method, and the feature extraction is performed outside the leading classification algorithm. This characteristic of the algorithm allows the system to be modified using state-of-the-art feature extraction algorithms.

Furthermore, the Cascaded-ANFIS algorithm is a combination of multiple Fuzzy Inference Systems. Therefore, it can synthesize and infer good combinations automatically. Therefore, implementing the algorithm by distinct fuzzy reasoning methods can generate optimized solutions. The CNN-based processes operate as a black box, and the alternations of the functions can be challenging. Therefore, the Cascaded-ANFIS algorithm has many merits over the traditional CNNs.

3.5 Conclusion

The fruit-360 dataset has 131 fruit classes with 90483 sample images, and many researchers tried to classify fruits in the dataset using artificial intelligence and machine learning techniques. However, none of the previous attempts focused on handling all 131 fruit classes with a total number of fruit images. Therefore, a novel and successful attempt are presented in this research work to identify all images in the Fruit-360 dataset using a Cascaded-ANFIS algorithm. The capability in image-based classification performance of the Cascaded-ANFIS algorithm was tested using nine feature descriptors. Thus, a robust and comprehensive Cascaded-ANFIS algorithm is presented in this research work.

The performance of the tested algorithm was tested using the learning curve and the confusion matrix. It can be concluded herein that the Cascaded-ANFIS algorithm outperformed all other state-of-the-art algorithms available for the specific task. The weighted precision, recall, and FScore reached their highest accuracies at 0.9843, 0.9841, and 0.9840, respectively, for the unbalanced Fruit-360 dataset. Therefore, the results provide compelling evidence that the Cascaded-ANFIS algorithm can handle multiple class image classification problems with higher cost-effectiveness and comparative accuracy than the CNN-based methods in past studies.

In addition, the algorithm showcased its capacities and capabilities in handling the total Fruit-360 data set at lower computational power. According to the results, it can be concluded that the Cascaded-ANFIS-based classifiers are suitable for real-time and cost-effective system implementations. The Cascaded-ANFIS architecture is an automatic cascade connection to the truth space approach of FIS. Therefore, Cascaded-ANFIS can rinse off the approximate reasoning part and make the reasoning of primary elements. Moreover, the interaction selection of Cascaded-ANFIS works as the best choice with FIS. A significant limitation of using the Cascaded-ANFIS algorithm is that it may need a different structure to obtain better accuracy for each dataset, such as a specific number of levels and a total number of inputs.

References

[1] Namal Rathnayake, Upaka Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. An efficient automatic fruit-360 image identification and recognition using a novel modified cascaded-anfis algorithm. *Sensors*, 22(12):4401, 2022.

[2] Baohua Zhang, Wenqian Huang, Jiangbo Li, Chunjiang Zhao, Shuxiang Fan, Jitao Wu, and Chengliang Liu. Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review. *Food Research International*, 62:326–343, 2014.

[3] Yudong Zhang, Lenan Wu, Shuihua Wang, and Genlin Ji. Comment on 'principles, developments and applications of computer vision for external quality inspection of

fruits and vegetables: A review (food research international; 2014, 62: 326–343)'. *Food Research International*, 70:142, 2015.

[4] Jean AT Pennington and Rachel A Fisher. Classification of fruits and vegetables. *Journal of Food Composition and Analysis*, 22:S23–S31, 2009.

[5] Anderson Rocha, Daniel C Hauagge, Jacques Wainer, and Siome Goldenstein. Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture*, 70(1):96–104, 2010.

[6] Xinwei Feng, Qinghua Zhang, and Zhongliang Zhu. Rapid classification of citrus fruits based on raman spectroscopy and pattern recognition techniques. *Food Science and Technology Research*, 19(6):1077–1084, 2013.

[7] P Cano Marchal, D Martínez Gila, J Gámez García, and J Gómez Ortega. Expert system based on computer vision to estimate the content of impurities in olive oil samples. *Journal of Food Engineering*, 119(2):220–228, 2013.

[8] Ruud M Bolle, Jonathan H Connell, Norman Haas, Rakesh Mohan, and Gabriel Taubin. Veggievision: A produce recognition system. In *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*, pages 244–251. IEEE, 1996.

[9] Woo Chaw Seng and Seyed Hadi Mirisaee. A new method for fruits recognition system. In *2009 International conference on electrical engineering and informatics*, volume 1, pages 130–134. IEEE, 2009.

[10] Shuihua Wang, Yudong Zhang, Genlin Ji, Jiquan Yang, Jianguo Wu, and Ling Wei. Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic abc and biogeography-based optimization. *Entropy*, 17(8): 5711–5728, 2015.

[11] T Pholpho, S Pathaveerat, and P Sirisomboon. Classification of longan fruit bruising using visible spectroscopy. *Journal of Food Engineering*, 104(1):169–172, 2011.

[12] Ce Yang, Won Suk Lee, and Jeffrey G Williamson. Classification of blueberry fruit and leaves based on spectral signatures. *Biosystems engineering*, 113(4):351–362, 2012.

[13] Yudong Zhang and Lenan Wu. Classification of fruits using computer vision and a multiclass support vector machine. *sensors*, 12(9):12489–12505, 2012.

[14] Mohammed Abdulraheem Fadhel, Ahmed Samit Hatem, Muhanad Abdul Elah Alkhalisy, Fouad H Awad, and Laith Alzubaidi. Recognition of the unripe strawberry by using color segmentation techniques. *International Journal of Engineering & Technology*, 7(4):3383–3387, 2018.

[15] Eduardo Garcia Breijo, Valeria Guarrasi, Rafael Masot Peris, Miguel Alcañiz Fillol, and Cristian Olguin Pinatti. Odour sampling system with modifiable parameters applied to fruit classification. *Journal of Food Engineering*, 116(2):277–285, 2013.

[16] FH Fan, Q Ma, J Ge, QY Peng, William W Riley, and SZ Tang. Prediction of texture characteristics from extrusion food surface images using a computer vision system and artificial neural networks. *Journal of Food Engineering*, 118(4):426–433, 2013.

[17] Mahmoud Omid, Mahmoud Soltani, Mohammad Hadi Dehrouyeh, Seyed Saeid Mohtasebi, and Hojat Ahmadi. An expert egg grading system based on machine vision and artificial intelligence techniques. *Journal of food engineering*, 118(1): 70–77, 2013.

[18] Yudong Zhang, Shuihua Wang, Genlin Ji, and Preetha Phillips. Fruit classification using computer vision and feedforward neural network. *Journal of Food Engineering*, 143:167–177, 2014.

[19] M Khanmohammadi, F Karami, A Mir-Marqués, A Bagheri Garmarudi, S Garrigues, and M De La Guardia. Classification of persimmon fruit origin by near infrared spectrometry and least squares-support vector machines. *Journal of food engineering*, 142:17–22, 2014.

[20] Supawadee Chaivivatrakul and Matthew N Dailey. Texture-based fruit detection. *Precision Agriculture*, 15(6):662–683, 2014.

[21] Ghulam Muhammad. Date fruits classification using texture descriptors and shape-size features. *Engineering Applications of Artificial Intelligence*, 37:361–367, 2015.

[22] Raheel Siddiqi. Effectiveness of transfer learning and fine tuning in automated fruit image classification. In *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies*, pages 91–100, 2019.

[23] Ghazanfar Latif, Batool Alsalem, Wejdan Mubarky, Nazeeruddin Mohammad, and Jaafar Alghazo. Automatic fruits calories estimation through convolutional neural networks. In *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, pages 17–21, 2020.

[24] Sourodip Ghosh, Md Jashim Mondal, Sourish Sen, Soham Chatterjee, Nilanjan Kar Roy, and Suprava Patnaik. A novel approach to detect and classify fruits using shufflenet v2. In *2020 IEEE Applied Signal Processing Conference (ASPCON)*, pages 163–167. IEEE, 2020.

[25] Seda Postalcıoğlu. Performance analysis of different optimizers for deep learning-based image recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(02):2051003, 2020.

[26] Ziliang Huang, Yan Cao, and Tianbao Wang. Transfer learning with efficient convolutional neural networks for fruit recognition. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 358–362. IEEE, 2019.

[27] Namal Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. A novel optimization algorithm: Cascaded adaptive neuro-fuzzy inference system. *International Journal of Fuzzy Systems*, pages 1–17, 2021.

[28] Horea Mureşan and Mihai Oltean. Fruit recognition from images using deep learning. *arXiv preprint arXiv:1712.00580*, 2017.

[29] Joseph C Harsanyi and C-I Chang. Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach. *IEEE Transactions on geoscience and remote sensing*, 32(4):779–785, 1994.

[30] Tohru Ueda and Yoko Hoshiai. Application of principal component analysis for parsimonious summarization of dea inputs and/or outputs. *Journal of the Operations Research Society of Japan*, 40(4):466–478, 1997.

[31] Dingwen Zhang, Junwei Han, Long Zhao, and Deyu Meng. Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework. *International Journal of Computer Vision*, 127(4): 363–380, 2019.

[32] Warren S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[33] J Paul Siebert. Vehicle recognition using rule based methods. 1987.

[34] William H Wolberg and Olvi L Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the national academy of sciences*, 87(23):9193–9196, 1990.

[35] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89 (1-2):31–71, 1997.

[36] Bangalore S Manjunath, J-R Ohm, Vinod V Vasudevan, and Akio Yamada. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11(6):703–715, 2001.

[37] Ka-Man Wong, Lai-Man Po, and Kwok-Wai Cheung. Dominant color structure descriptor for image retrieval. In *2007 IEEE International Conference on Image Processing*, volume 6, pages VI–365. IEEE, 2007.

[38] Dong Kwon Park, Yoon Seok Jeon, and Chee Sun Won. Efficient use of local edge histogram descriptor. In *Proceedings of the 2000 ACM workshops on Multimedia*, pages 51–54, 2000.

[39] Sawet Somnugpong and Kanokwan Khiewwan. Content-based image retrieval using a combination of color correlograms and edge direction histogram. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5. IEEE, 2016.

[40] C-H Lin, C-W Liu, and H-Y Chen. Image retrieval and classification using adaptive local binary patterns based on texture features. *IET image processing*, 6(7):822–830, 2012.

[41] Mryka Hall-Beyer. Glcm texture: a tutorial. *National Council on Geographic Information and Analysis Remote Sensing Core Curriculum*, 3:75, 2000.

[42] G Lowe. Sift-the scale invariant feature transform. *Int. J*, 2(91-110):2, 2004.

[43] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[44] N Gayathri and K Mahesh. Improved fuzzy-based svm classification system using feature extraction for video indexing and retrieval. *International Journal of Fuzzy Systems*, 22:1716–1729, 2020.

[45] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[46] Paul L Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, 1999.

[47] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[48] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[49] Jessica D Bayliss, J Anthony Gualtieri, and Robert F Cromp. Analyzing hyperspectral data with independent component analysis. In *26th AIPR Workshop: Exploiting New Image Sources and Sensors*, volume 3240, pages 133–143. International Society for Optics and Photonics, 1998.

Chapter 4

Hydropower forecasting in Sri Lanka

Chapter 4 focuses on an application developed to solve a real-world problem. This study was conducted to predict and forecast the hydropower generation in Samanalawewa Reservoir in Sri Lanka. The results of this study show that the Cascaded-ANFIS can handle regression problems. This work is published in Sensors (MDPI) [1].

4.1 Introduction

The Sustainable Development Targets (SDGs) were announced in 2012, with 17 goals recommended for completion by 2030. One of the essential aims of the list is to achieve clean energy from power generation [2]. Global hydropower output peaked in 2020 with 38.2 exajoules, up from 37.7 exajoules the previous year, and climbed by 11.6 exajoules in the two decades from 2000 to 2020 [3]. Thus, hydropower contributes to more than 16% of total energy generation [4]. Many South Asian nations, including Sri Lanka, fulfil a considerable portion of their electrical demand through hydropower facilities (approximately 40% of total energy in Sri Lanka) [5]. Renewables are still regarded as one of the world's most environmentally friendly power-producing systems. As a result, a 75—100% increase in production capacity is projected in the coming years [4]. Compared to wealthy countries, which have utilized 70% of their capacity, emerging nations have only evaluated 23% of financially feasible hydropower plants [6]. As a result, many developing nations are rapidly spending considerable resources on developing hydropower facilities since it is seen as a safe and cost-effective source of renewable energy that minimizes carbon emissions [7].

Along these lines, hydropower is one of the cleanest forms of energy sources; however, the inflow to dam reservoirs significantly impacts the pace of hydropower output. Therefore, hydropower generation, on the other hand, is very unpredictable due to its dependency on meteorological and weather conditions. Furthermore, climate change will likely disrupt hydropower plant operations by unbalancing the water cycle, increasing the frequency of rainfall events, and rising atmospheric temperatures. It is evident that the evaporation and other water cycle components are affected by the predicted temperature change of 0.0164 °C annually [8]. Rainfall, on the other hand, is projected to increase in some countries while decreasing in others, thus impacting hydropower-producing capacity [9].

If electricity output is dramatically curtailed due to climate change's negative consequences of climate change, the hydropower sector might become one of the most vulnerable businesses. In addition, water scarcity in the catchment and reduced hydropower generation inputs due to landslides or soil erosion might exacerbate the problem. On the other hand, the construction of hydroelectric infrastructure is prohibitively expensive, presents substantial dangers to the aquatic ecology, and produces socioeconomic concerns [10].

As a result, forecasting hydropower output is critical for maximizing renewable energy consumption to meet growing demand and control hydroelectric power management. This will help to achieve environmental sustainability. Despite this, estimating future hydropower output is challenging due to the nonlinearities of the input functions and regional and temporal fluctuations in meteorological data, including temperature and rainfall. As a result, the prediction output of the model might have a substantial financial benefit in regulating renewable energy infrastructure development like hydroelectric [11].

4.2 Related Works

Several researchers have studied the impact of climatic fluctuation on hydroelectric output, primarily utilizing Global/Regional Climate Models (GCMs/RCMs), predictive modelling, and conventional statistical methodologies (e.g., [12–14]).

Several methods to predict the future of hydropower plants using machine learning techniques can be found in the literature, and ANN is one of the main algorithms that can be used to carry out this task. A case study was carried out in Nigeria, as well as in Jebba and Kainji, employing ANN impartial input data [15]. In Uzlu et al.[16], the artificial bee colony method was used to forecast future hydropower output throughout Turkey, utilizing input factors including generation capacity, energy consumption, population, and temperatures. According to the report, the Power output of Turkey is not in accordance with the country's objective of producing 30% of its renewable electricity in 2023. Furthermore, Patil [17] examined future streamflow for the Ranganadi River, which is located in India up to 2040, to forecast hydropower output using three GCM models and ANN. When using feed-forward back-propagation algorithms on the ANN architecture, input parameter characteristics substantially influence forecasting future power generation [18].

Furthermore, while projecting electricity output from various energy resources in the United States, Khodaverdi [19] proposed an ANN-ARIMA hybrid model rather than ANN to predict future renewable energy resources data (e.g., hydroelectricity, solar, and wind). After examining 66 studies that used ANN to improve reservoir operations, the study by Ajala et al. [20] further reinforced the idea of combining ANN with supervised or unsupervised learning algorithms to improve reservoir outflow prediction. Furthermore, the study by Shaktawat and Vadhera [6] advised further research on risk management in hydropower utilizing a fuzzy model mixed with ANN and genetic algorithm.

Some scientists insist that ANNs are important in hydropower prediction. Anuar et al. have showcased that the hidden layer neurons had a more significant impact on the results of the ANN structure when forecasting streamflow at The Malaysian hydroelectric dam [21]. Furthermore, Sessa et al. [22] discovered that ANN models are the most accurate in predicting short-term and long-term hydropower generation after having conducted research studies in run-of-the-river (ROR) hydroelectricity in France, Portugal, and Spain using chronological weather information such as rainfall, snow, and temperature.

However, the related research in the context of Sri Lanka is minimum. In fact, as per the authors' knowledge, only one such research was available in Sri Lanka that used ANN to anticipate electricity output. Furthermore, the research by Karunathilake and Nagaha [23] estimated daily electricity consumption but did not forecast power generation.

Although numerous ANN-based machine learning algorithms have been found in the literature for hydropower prediction, machine learning techniques that use Fuzzy Logic to predict hydropower generation are a handful. Some of the literature on Fuzzy Logic-based predictions can be listed as follows.

The Grey wolf approach was combined with an adaptive neuro-fuzzy inference system (ANFIS) to anticipate hydroelectricity generation Dehghani et al. [24]. In addition, the hydropower output of Albania was analyzed by Konica, and Staka [25] to establish the best forecasting model for assessing hydro energy production for the years 2007-2016. They have used the fuzzy time series approach to forecast Albania's hydropower generation.

Moreover, some studies have been conducted to forecast rainfall using Fuzzy Logic based algorithms. The rainfall forecast is done in this study in a study by Suprapty et al. [26] in the East Kalimantan area, which has 13 watersheds with the potential to build a Micro Hydro Power Plant. The Auto-Regressive (AR) Model based on the Fuzzy Inference System (FIS) is utilized to simulate rainfall time series data. The research work done by Rahman et al. [27] has showcased an improvement in forecasting rainfall using a fuzzy rule-based approach. Eight different equations have been created using temperature, wind velocity, and precipitation. The minimum content of the induction component of temperature and wind velocity fuzzifications is investigated, as are fuzzy levels and membership functions.

Mostly, time-series predictions are purely non-linear, and fuzzy logic is the best in artificial intelligence to tackle problems in non-linear [28].

The majority of the earlier works share the following flaws.

1. Generally, Artificial Neural Network-based algorithms are bulky in the complexity of the calculations.

2. Difficult to use when the predictions depend on the uncertainty factors and non-linear inputs.

3. It is not likely to generate the best possible prediction because the input factors vary depending on the different environments.

4. Requires an enormous amount of computing power.

Therefore, while addressing the above-mentioned overall flaws, this study presents a new algorithm called Cascaded Adaptive Neuro-Fuzzy Inference System (Cascaded ANFIS) to predict the hydropower generation [29]. The impact of this research can be pointed out as follows.

1. This system uses fuzzy logic approach along with Neural Network to address the inputs' uncertainty and non-linearity.

2. Since the base algorithm of this system is two-input one-output ANFIS, the computational power reduces dramatically.

3. It is possible to generate a near-zero error in the prediction by increasing the number of levels in the Cascaded ANFIS algorithm.

4. This study presents future power generation up to the year 2099 in two different climate models.

5. The comparative study presented in this work provides a solid understanding of the potential regarding the Cascaded ANFIS algorithm upon the cutting-edge time series prediction algorithms.

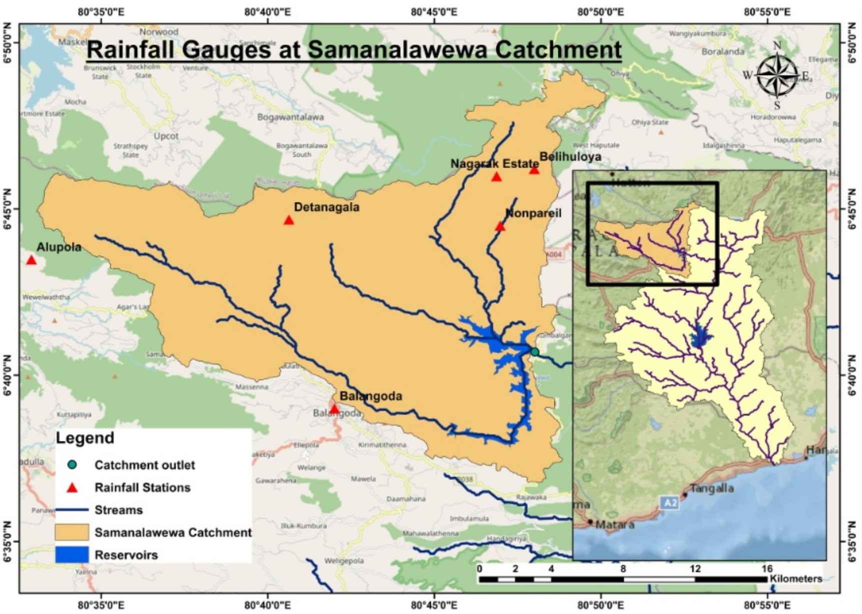


Figure 4.1: Rainfall Gauges at Samanalawewa catchment

4.2.1 Hydropower in Sri Lanka

Sri Lanka has a hydroelectric power potential of 1,719 Megawatts (MW), and existing hydropower growth pledges would contribute around 247 MW to the power grid mostly in coming decades [5]. According to Gunasekara [30], the bulk of Sri Lanka's hydroelectric plants are more than 25 years old. Although hydropower plants have a lifespan of about 50 years, if any of the older hydroelectric dams fail to operate, whether due to climate

or mechanical fault, Sri Lanka will then meet an energy shortage problem because it will be challenging to replace the defective hydroelectric dams in a brief period [31].
As a result, analyzing the power-generating capabilities of hydroelectric projects in the Sri Lankan context is crucial. To handle a developing country's economic electrical demands and manage water supply infrastructural development amid climatic factors. However, several analyses in Sri Lanka have looked at potential energy production from current or planned hydroelectric dams. The study in Udayakumara et al. [32] looked at ways to increase power output in hydroelectric dams by preventing land degradation and reservoir floods in the Uma Oya valley, one of Sri Lanka's most crucial significant catchment areas.
The study Chandrasekara et al. [3] studied inflows in the Kotmale reservoir until 2005 from 1960 using the El Nino Southern Oscillation (ENSO) phase indicator and discovered that flow to the basin had decreased, impacting hydropower output and agricultural plans. According to the research in Imbulana et al. [33], a rise in continuous rainfall events, a decrease in continuous dry weather, and a gain in yearly rainfall series will improve the future production capacity of the Mahaweli watershed's hydropower plants. In addition, Khaniya et al. [13] used a multiyear rainfall trend research to demonstrate that climate changes will not affect Denawaka Ganga mini-hydropower generating as in the Rathnapura area. The study released in Perera and Rathnayake [34] additionally sought to analyze the effect of climate change on the Erathna mini-hydropower station in the Rathnapura area. They concluded that electricity generation would decline in the following years.
The study by Khaniya et al. [31] [35] undertook a similar evaluation of the recently operational Uma Oya watershed, and the researchers found that there will be no substantial challenges to hydroelectric generation in the years ahead groundwater limits in the watershed region. Nevertheless, as stated in the introduction, there seems to be no comprehensive study on hydroelectric forecasts in Sri Lanka for the coming decades. Consequently, this study can better attract the attention of the Sri Lankan authorities to enhance the management and forecasting procedures in hydroelectric plants.

4.3 Study Area

The Samanalawewa Hydropower Project is located in the central portion of Sri Lanka, in the Belihul Oya region of Rathnapura division, Sabaragamuwa province. The project was completed in 1992, just downstream of the confluence of Belihul Oya to Walawe River. The watershed region (359 km2) is midland, made of marble and quartz, and has an average altitude of around 530 m [31]. The region is located inside the rainy region of the country (wet zone), with a mean annual precipitation of around 2500 mm [36]. The southwest monsoon provides the majority of the rainfall for the catchment, with minor contributions from the northeast monsoon and inter-monsoon storms. The Samanalawewa Hydroelectric power project includes a U-shaped rockfill dam around 110 m high from its foundation. The power station is capable of producing 124 MW as per the design guidelines. Figure 1 illustrates a detailed catchment map.

Samanalawewa hydroelectric is among Sri Lanka's oldest and largest reservoir-type power stations and has long played an essential part in maintaining power distribution stability during peak times. It accounts for 8.69% of all extensive hydroelectric plants providing electricity for Sri Lanka's electrical requirements. Since its start, this project has aroused significant attention owing to the leakage problem discovered on the lake's right bank due to poor geological characteristics [37]. Moreover, several environmental difficulties were noted during the design stage; however, little awareness was taken because no stringent environmental restrictions necessitated substantial development efforts [38].

Although the Environmental Impact Assessment (EIA) framework was established in Sri Lanka in 1988, EIA during the building of Samanalawewa was primarily centred on vegetation revascularization and habitat conservation.

Due to the apparent leak, phase 2 of the construction of the hydropower plant (120 MW capacity) was suspended; therefore, a mini-hydropower facility was constructed to utilise the leaking water. Despite the Ceylon Electricity Board's (CEB's) valiant efforts to halt the leak, stored water continues to flow at a pace of 2.1–2.8 m^3/s [39].

Irrigated water from the dam is vital for agricultural usage in downstream settlements such as Kaltota, Madabadda, Welipotayaya, and Koongahamankada. Paddy yields downstream of the study area have been reduced by 11.5 per cent due to a lack of water in the reservoir [40]. Therefore, water management is highly important.

Because a portion of the confiscated water is immediately delivered for irrigation without going through the power station, analyzing the prospective availability of water in the Samanalawewa dam for energy production is crucial. Another fraction (the leaking component) is supplied by mini-hydropower plants that produce far less energy. Furthermore, water management at the Samanalawewa reservoir must be more carefully managed with the rising availability of water from downstream agricultural districts. Furthermore, climate variability may influence CEB's watershed management goals at the Samanalawewa hydroelectric station, either positively or negatively. As a result, the following study will interest the many stakeholders of the Samanalawewa Hydropower Project.

To assess that, the monthly rainfall data were purchased from the Department of Meteorology, Sri Lanka, for the rainfall stations showcased in Figure 4.1. The data was collected from 1992 to 2018 as per the availability. There were some missing data due to various reasons, including instrumentation errors. Therefore, the data were screened carefully before they were used. Balangoda, Alupola, Detanagalla, Belihuloya, Nonpareil (Belehuloya), and Nagrak Estate are the six stations used in this study.

4.4 Methodology

The overall explanation of the method used in this study is presented in this section. The development process is several steps. Initially, futuristic climate data were extracted and corrected their biases using the linear bias correction technique. Then the Cascaded ANFIS algorithm is used to generate the outputs for each pair of inputs. This process is explained in the algorithm usage subsection.

Furthermore, three state-of-the-art algorithms, GRU, RNN, and LSTM, are used to distinguish the efficiency of the algorithms.

4.4.1 Climate data extraction for future

Global Climatic Models (GCMs) accommodate climatic data at vast ranges across immensely different landscapes. In contrast, Regional Climatic Models (RCMs) are employed at more inadequate orders and can accommodate more specific data for adaptation evaluation and preparation [41]. As a projected instrument, GCMs forecast the climate variance of the Earth in the future. They should, however, be investigated on a local or even global scale to identify efficient correspondence procedures.

Future climatic data for various situations can be retrieved. Such scenarios are known as Representative Concentration Pathways (RCP), where weather data can be obtained. RCPs can be expressed as trajectories on the Intergovernmental Panel on Climate Change's [42] greenhouse gas concentrations. RCP 2.6, 4.5, 6.0, and 8.5 are the four most generally applied RCPs in the literature [42]. RCP4.5 is the intermediate emission scenario, in which emissions begin to decline around 2045, where RCP8.5 is the leading emission situation, in which discharges proceed to rise during the 21^{st} century.

It is generally known that RCMs have variable degrees of methodical bias [43, 44]. The causes of such preferences could be due to methodical model mistakes produced through poor conceptualizations, spatial averaging, and discretizations in grid cells. Some prejudice improvement strategies are employed in the literature to address these biases [45]. Linear scaling, local intensity scaling, power transformation, variance scaling, distribution transfer, and delta change approach are widely used techniques in removing biases in climatic data.

The Linear Scaling (LS) approach [46] is employed extensively in various investigations due to its simplicity and speed of application. LS can adjust all-climate elements to an appropriate level; however, a few examples of precipitation corrections can be found in Gimire et al., Lafon et al., Luo et al., and Mahmood et al. [47–50]. The bias correction method for linear scaling can be implemented employing the two equations provided here (Equations (4.1) and (4.2)), where *his*, *cor*, *sim*, *obs*, *d*, and *P* stand for raw RCM data, bias-corrected data, raw RCM corrected data, observed data, daily, and precipitation, respectively, and m is the long-term cyclical average of rainfall data:

$$P^{cor}_{his,d} = P_{his,d} * \frac{\mu_m(P_{obs,d})}{\mu_m(P_{his,d})} \tag{4.1}$$

$$P^{cor}_{sim,d} = P_{sim,d} * \frac{\mu_m(P_{obs,d})}{\mu_m(P_{sim,d})} \tag{4.2}$$

LS technique was used to remove the biases in the RCP precipitation products as shown in the Equations 4.1 and 4.2. The ground-measured monthly rainfalls were used to remove these biases.

As mentioned in the above sections on dataset generation for future rainfall, four data points are generated for every month in the range from the year 2021 to the year 2099 using RCP 4.5 and RCP 8.5 climate models. Accordingly, these four data points were used as the inputs to the Cascaded ANFIS algorithm. In the end, the mean of the outputs is calculated as the final solution f (equation 4.3). Where, $O_{n,j}$ is the output of n^{th} level j^{th} node.

$$f = \frac{\sum_{j=1}^{4} O_{n,j}}{4} \tag{4.3}$$

4.4.1.1 Parameter settings for each algorithm

This study is conducted to investigate the best prediction algorithm from the state-of-the-art algorithms in hydropower forecasting. Hence, there are several algorithms used, and each algorithm is created with the optimum parameters. Following is the complete list of algorithms used in this study.

1. Multilayer Perception (MLP)
2. K - Nearest Neighbors (KNN)
3. Adaptive Network-based Fuzzy Inference System (ANFIS)
4. Particle Swarm Optimization with ANFIS (ANFIS-PSO (Hybrid))
5. Genetic Algorithms with ANFIS (ANFIS-GA (Hybrid))
6. Linear Regression
7. Lasso Regression
8. Ridge Regression
9. Recurrent Neural Network (RNN)
10. Long Short-Term Memory (LSTM)
11. Gated Recurrent Unit (GRU)
12. Cascaded ANFIS

Here, two algorithms were used: general machine learning algorithms and regression machine learning algorithms. MLP, KNN, and ANFIS methods can be presented as general machine learning algorithms, while Linear, Lasso, Ridge, LSTM, GRU, and RNN can be introduced as regression models.
Each algorithm is separately coded and run during the study to generate the outputs. Most of the algorithm parameters are manually adjusted, while some of the algorithms are adjusted under the consideration of literature studies. Each parameter for each algorithm is shown in Table 10.1.

Table 4.1: Parameter Setting for each algorithm

Algorithm	Parameters	
MLP	Hidden layer size	50, 50, 50
	Activation	tanh
	Solver	adam
	alpha	0.05
	learning rate	constant
KNN	Weights	Uniform
	n_neighbors	1
ANFIS	Iteration	100
	Membership Functions	3
	Step Size	0.1
	Decrease rate	0.9
	Increase rate	1.1
ANFIS-PSO	Inertia Weight	1
	Inertia weight damping ratio	0.99
	Personal Learning Coefficient	1
	Global Learning Coefficient	2
ANFIS-GA	Crossover Percentage	0.7
	Mutation Percentage	0.5
	Mutation Rate	0.1
	Selection Pressure	8
	Gamma	0.2
RNN / LSTM / GRU	Optimizer	adam

Table 4.1: Parameter Setting for each algorithm

Algorithm	Parameters	
	Learning rate	0.0001
	Activation	relu
	batch size	30
	epochs	100
Cascaded ANFIS	Iteration	100
	Membership Functions	3
	Step Size	0.1
	Decrease rate	0.9
	Increase rate	1.1

The experiment was carried out for the hydropower generation dataset. Nine different algorithms were tested, and the best algorithm was chosen based on the Root Mean Square Error ($RMSE$) and the Coefficient of Determination (R^2) of each algorithm. The RMSE and R^2 can be calculated as shown in Equation 4.4,4.5.

$$RMSE = \sqrt{\frac{1}{q}\sum_{t=1}^{q}(\bar{u}(t) - \hat{u}(t))^2} \tag{4.4}$$

$$R^2 = 1 - \frac{RSS}{TSS} \tag{4.5}$$

Where, in Equation 4.4, $\bar{u}(t)$ is introduced as the prediction and $\hat{u}(t)$ is the real output. q is the size of the population. In Equation 4.5, the sum of the squares of the prediction is RSS and the sum of squares of real values is TSS.

4.5 Results and Discussion

This section includes two main subsections. First, algorithm comparison is introduced since selecting the best algorithm is one of the main objectives of this study. Second, the future power generation is explained along with the results with the best algorithm selected here.

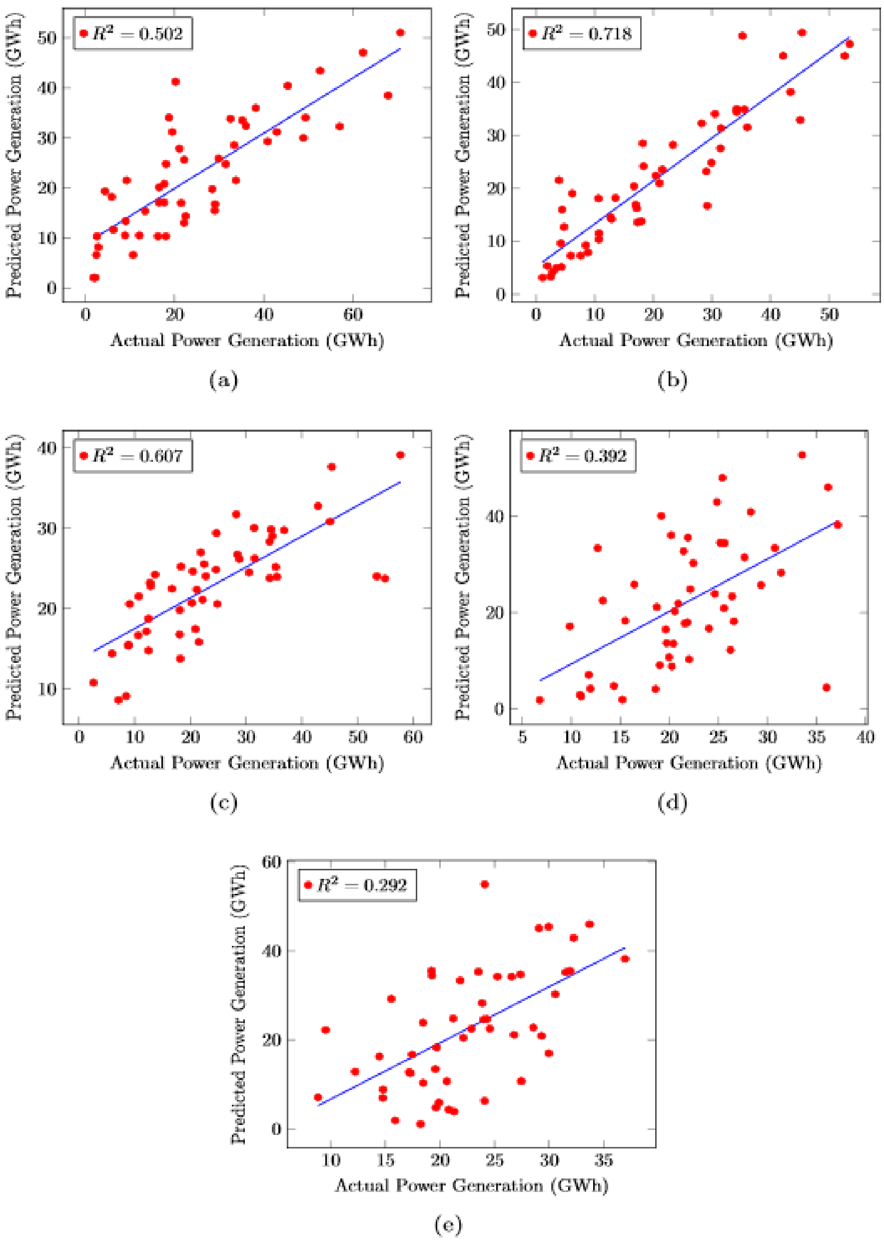


Figure 4.2: Coefficient of Determination (R^2) of Rain Fall Test dataset for (a) KNN, (b) MLP, (c) ANFIS (d) PSO-ANFIS and (e) GA-ANFIS

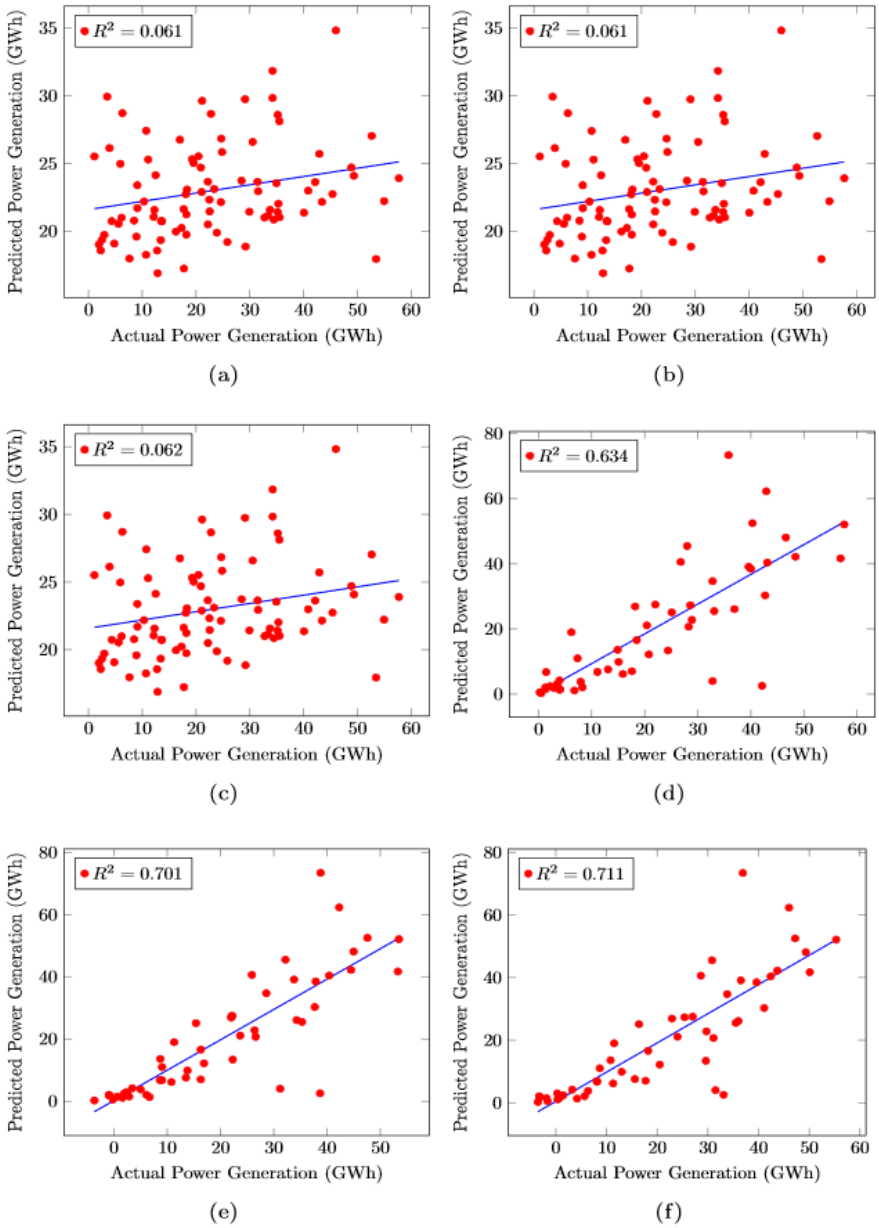


Figure 4.3: Coefficient of Determination (R^2) of Rain Fall Test dataset for (a) Linear Regression, (b) Lasso Regression, (c) Ridge regression (d) RNN, (e) LSTM and (f) GRU

4.5.1 Comparison of the Algorithms

Table 4.2 presents the RMSE for each algorithm at the training and testing phases. The slightest error of 1.01 in the training and 1.80 in the testing was obtained by the Cascaded ANFIS. As mentioned in the introduction of the Cascaded ANFIS, the error reduces while propagating through levels. Hence, a higher level of structure generates more accurate results at the cost of computational power. However, the results are for the Cascaded ANFIS at level 20.

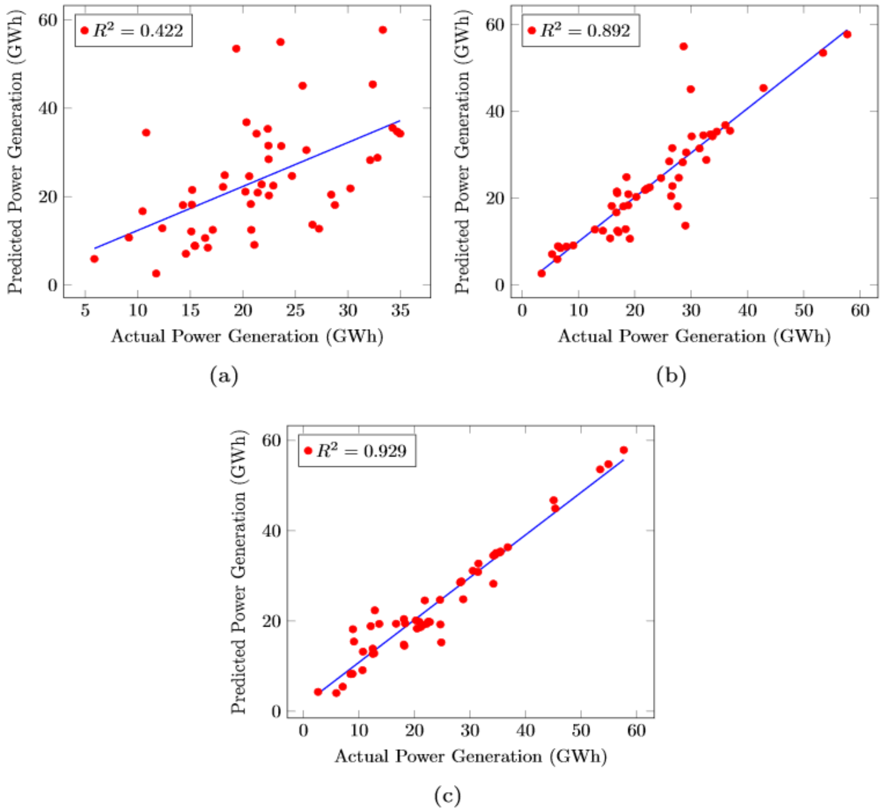


Figure 4.4: Cascaded ANFIS behavior for different levels.(a) Level 1, (b)Level 10, (c) Level 20

Moreover, the second, third, and fourth best accuracies are LSTM, GRU, and RNN. They obtained 6.03, 6.50, and 7.85 errors during the training sequentially. It is also worth remarking that the other ANFIS algorithms, such as ANFIS, ANFIS-PSO, and ANFIS-GA, present a higher error rate when compared with the other algorithms.

Furthermore, the Coefficient of Determination (R^2) is calculated for each algorithm as shown in Figures 4.2 and 4.3. Here, Figure 4.2 shows the performances of general machine learning algorithms and Figure 4.3 shows regression machine learning algorithm performances. R^2 is used to examine how variations in one variable may be explained by changes in another.

R^2 calculates the percentage variance in y explained by x-variables. The measure runs from 0 to 1. (the x-variables can explain, i.e. 0% to 100% of the variation in y).

The best R^2 is given by the Cascaded ANFIS as 0.929. while GRU, LSTM, and RNN calculate it as 0.711, 0.701, and 0.634, respectively.

The increase of R^2 of the Cascaded ANFIS by level can be seen in Figure 4.4. At level 1, R^2 is 0.422 because only two variables are considered the input to ANFIS modules at the first level. Then at level 10, the R^2 value has increased by almost 50%. Finally,

Table 4.2: RMSE for training and testing

Algorithm	RMSE (Train)	RMSE (Test)
MLP	7.52	25.26
KNN	9.73	19.33
ANFIS	10.47	18.06
ANFIS-PSO	10.99	16.61
ANFIS-GA	11.88	16.87
Linear Regression	13.74	14.85
Lasso Regression	13.72	14.82
Ridge Regression	13.70	14.88
RNN	7.85	11.62
GRU	6.50	8.33
LSTM	6.03	6.88
Cascaded ANFIS	1.01	1.80

at level 20, the value has reached almost 1 (0.929). Therefore this result explains that the Cascaded ANFIS algorithm outperforms all other algorithms, including regression models. Hence, the Cascaded ANFIS algorithm forecasts hydropower generation up to the year 2099.

4.5.2 Forecasting of Hydropower Generation for future

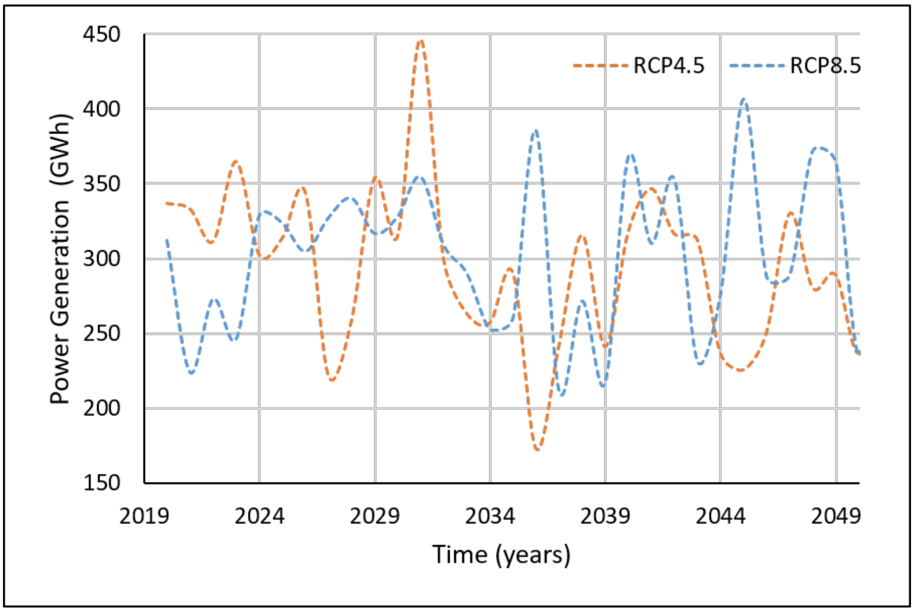


Figure 4.5: Hydropower Predictions from Khaniya et al (2020) [13]

Figure 4.6 showcases the projected power generation for the near future under the RCP4.5 and RCP8.5 climate scenarios. It can be seen herein that both climate scenarios have projected a significant decline in power generation in the Samanalawewa

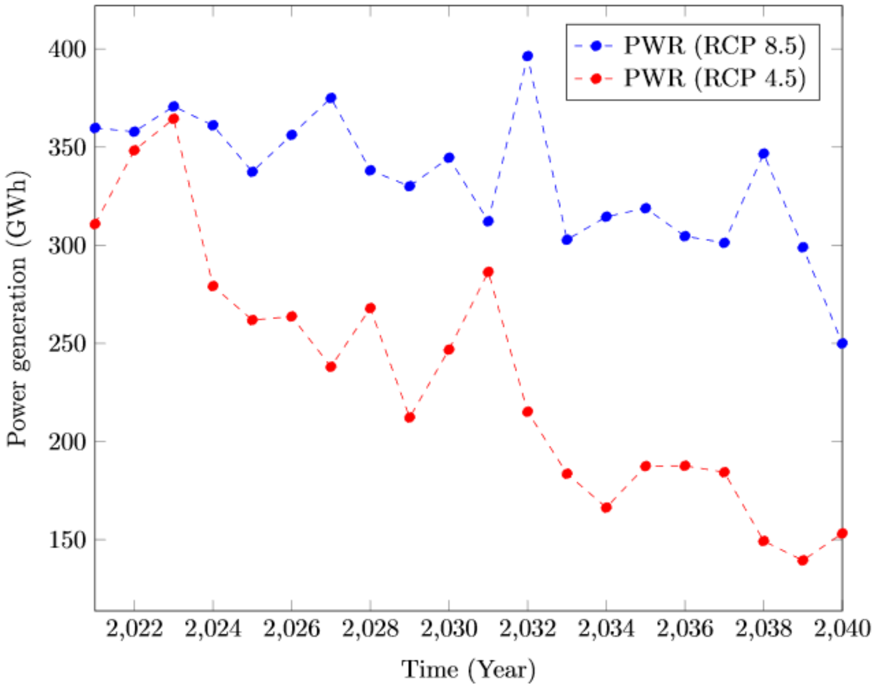


Figure 4.6: Power generation prediction from the year 2021 to 2040

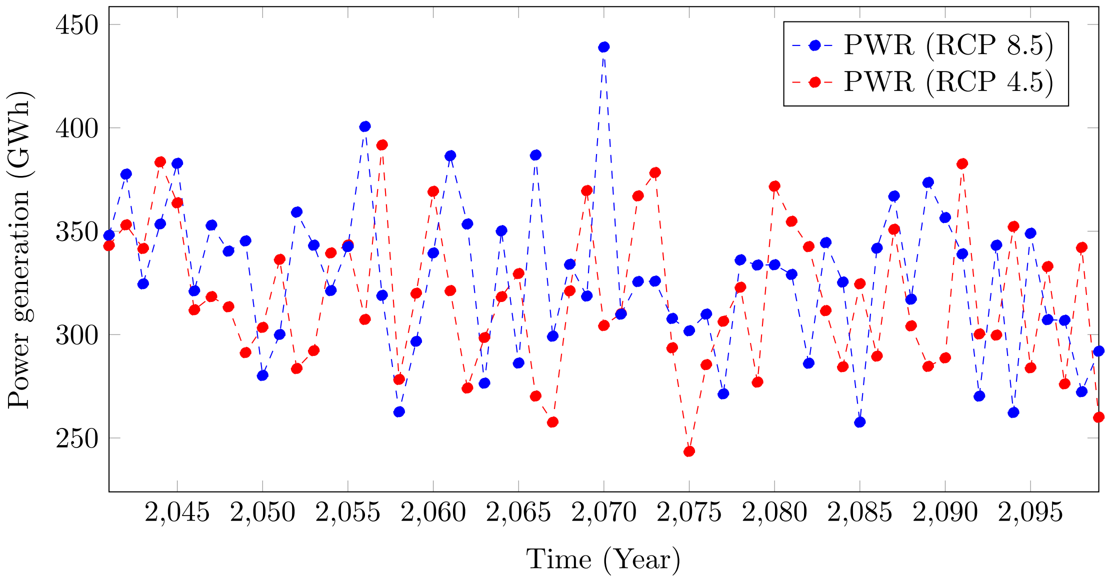


Figure 4.7: Power generation prediction from the year 2041 to 2099

Hydropower plant. The declination is monotonic except for a couple of years of slight inclinations. However, interestingly, the power generation in RCP4.5 is lower than that of RCP8.5. Many development projects are expected in Sri Lanka, requiring a significant power demand. It is projected around a 1000 MW power demand for Sri Lanka in the future. In addition, Sri Lanka has proposed to generate more than 70% of its power demand using renewable resources by the 2030s. However, the Samanalawewa power plant results for the near future do not support both requirements in the near future. This is critical as the power plant significantly contributes to Sri Lanka's power demand as a renewable resource.

Figure 4.7 presents the projected power generation for mid-future years (2041 to 2070) from both RCP scenarios. Unlike in the near future, the projected power generation patterns have zig-zag patterns for both climatic scenarios. However, they still showcase declining trends. In addition, the significant differentiation in the projected power generation from RCP4.5 and RCP8.5 for the near future cannot be seen in the mid-future. Instead, an overlap of both climatic scenarios can be seen.

Nevertheless, the projected power generations under RCP4.5 and RCP8.5 climatic scenarios showcase the impact of climate change on hydropower generation in a healthy hydropower plant in Sri Lanka. Even though Figures 4.6 and 4.7 present the annual power generations, seasonal impacts can also be seen on the higher resolution scales such as monthly power generations. Therefore, climate change is adversely impacting the Samanalawewa hydropower plant in the near future and mid-future, even though Sri Lanka's power demand is in escalating phase. Therefore, the findings of this research can be used for critical discussions by the stakeholders and then enhance the countermeasures.

Clear differences can be seen for the power generation prediction from two different techniques (Figure 4.5 and 4.6). Khaniya et al. (2020) [13] have frequently used ML algorithms in ANNs. Significant reductions can be seen for RCP4.5 under this Cascaded ANFIS algorithm. Therefore, the results have to be carefully assessed with time. The analysis can be restructured in the short term.

Figure 4.7 illustrates the projected power generation from 2041 to 2099. A similar illustration to mid-future (2041-2070) power generations can also be seen in the far future (2071-2099). However, the projections overall do not showcase declining or inclining trends, even though they have peaks and troughs. Nevertheless, as per the authors' understanding, it is too early to comment on power generation in the far future. RCP scenarios have projections for the far future; however, the high variability of climate and its relationship to greenhouse gas emissions might change future patterns. In addition, the world's green energy concepts, like electric vehicles, would positively impact the changing climates in the long run. Even though authors have found the projected power generations for the far future, quick conclusions may not be feasible.

4.6 Conclusion

Hydropower generation for the Samanalawewa hydropower plant was forecasted using a novel Cascaded ANFIS algorithm under RCP4.5 and RCP8.5 for future years. The newly utilised algorithms' accuracy is higher than other frequently used ones. It has shown lower RMSEs and higher R^2. The authorities would be interested in the prediction model due to its robustness for practical applications. However, the algorithm takes some significant time to train the forecasting model. The future projection is engaging. The projection was considered for the near future and mid-future cases based on the design life of a hydropower station. Therefore, the suggestions for future forecasting should align with the design life of the hydropower plant. Replacement of various essential instrumentation like turbines can significantly influence power generation efficiency. Therefore, the results presented herein are based on the currently available system. Based on these, the model can successfully be utilized to forecast power generation for future years. Thus, the authorities and planners can learn about the future generation and match the required demand. In addition, the authorities can make decisions regarding replacements of various instrumentation to enhance the efficiency of the Samanalawewa hydropower station.

References

[1] Namal Rathnayake, Upaka Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. A cascaded adaptive network-based fuzzy inference system for hydropower forecasting. *Sensors*, 22(8):2905, 2022.

[2] Ki-moon Ban. Sustainable development goals. 2016.

[3] Sewwandhi Chandrasekara, Venkatraman Prasanna, and Hyun-Han Kwon. Monitoring water resources over the kotmale reservoir in sri lanka using enso phases. *Adv. Meteorol.*, 2017, 2017.

[4] T Kober, H-W Schiffer, M Densing, and E Panos. Global energy perspectives to 2060–wec's world energy scenarios 2019. *Energy Strategy Rev.*, 31:100523, 2020.

[5] Asian Development Bank. *Tourism Sector Assessment, Strategy, and Road Map for Cambodia, Lao People's Democratic Republic, Myanmar, and Vietnam (2016-2018).* Asian Development Bank, 2017.

[6] Anuja Shaktawat and Shelly Vadhera. Risk management of hydropower projects for sustainable development: a review. *Environment, Development and Sustainability*, 23(1):45–76, 2021.

[7] Abid Hussain, Gopal K Sarangi, Anju Pandit, Sultan Ishaq, Nabir Mamnun, Bashir Ahmad, and Muhammad Khalid Jamil. Hydropower development in the hindu kush himalayan region: Issues, policies and opportunities. *Renew. Sust. Energ. Rev.*, 107: 446–461, 2019.

[8] Gong Chen and Gregory De Costa. Climate change impacts on water resources: case of sri lanka. 2017.

[9] G Naveendrakumar, Meththika Vithanage, Hyun-Han Kwon, MCM Iqbal, S Pathmarajah, and Jayantha Obeysekera. Five decadal trends in averages and extremes of rainfall and temperature in sri lanka. *Adv. Meteorol. ADV METEOROL*, 2018, 2018.

[10] Xiao Zhang, Hong-Yi Li, Zhiqun Daniel Deng, Claudia Ringler, Yang Gao, Mohamad I Hejazi, and L Ruby Leung. Impacts of climate change, policy and water-energy-food nexus on hydropower development. *Renew. Energy*, 116:827–834, 2018.

[11] Deepika Yadav, N Veena Sharma, et al. Artificial neural network based hydro electric generation modelling. *Int. J. Appl. Eng. Res.*, 2(1):56–72, 2010.

[12] Mohammad Sajjad Khan, Paulin Coulibaly, and Yonas Dibike. Uncertainty analysis of statistical downscaling methods. *J. Hydrol.*, 319(1-4):357–382, 2006.

[13] Bhabishya Khaniya, Harshana G Priyantha, Nilushi Baduge, Hazi Md Azamathulla, and Upaka Rathnayake. Impact of climate variability on hydropower generation: a case study from sri lanka. *ISH J. Hydraul. Eng.*, 26(3):301–309, 2020.

[14] Pengcheng Qin, Hongmei Xu, Min Liu, Liangmin Du, Chan Xiao, Lüliu Liu, and Brian Tarroja. Climate change impacts on three gorges reservoir impoundment and hydropower generation. *J. Hydrol.*, 580:123922, 2020.

[15] TS Abdulkadir, AW Salami, AR Anwar, and AG Kareem. Modelling of hydropower reservoir variables for energy generation: neural network approach. *EJESM*, 6(3): 310–316, 2013.

[16] Ergun Uzlu, Adem Akpınar, Hasan Tahsin Özturk, Sinan Nacar, and Murat Kankal. Estimates of hydroelectric generation using neural networks with the artificial bee colony algorithm for turkey. *ENEYDS*, 69:638–647, 2014.

[17] Manti Patil. Stream flow modeling for ranganadi hydropower project in india considering climate change. *Curr. World Environ.*, 11(3):834, 2016.

[18] Ali Thaeer Hammid, Mohd Herwan Bin Sulaiman, and Ahmed N Abdalla. Prediction of small hydropower plant power production in himreen lake dam (hld) using artificial neural network. *Alex. Eng. J.*, 57(1):211–221, 2018.

[19] Maryam Khodaverdi. Forecasting future energy production using hybrid artificial neural network and arima model. 2018.

[20] Abiodun Ladanu Ajala, Josiah Adeyemo, and Semiu Akanmu. The need for recurrent learning neural network and combine pareto differential algorithm for multi-objective optimization of real time reservoir operations. *J. Soft Comput. Civ. Eng.*, 4(3):52–64, 2020.

[21] N Anuar, M Khan, Jagadeesh Pasupuleti, and A Ramli. Flood risk prediction for a hydropower system using artificial neural network. *Int J Recent Technol Eng.*, 8 (4):6177–6181, 2019.

[22] Valentina Sessa, Edi Assoumou, and Mireille Bossy. Modeling the climate dependency of the run-of-river based hydro power generation using machine learning techniques: an application to french, portuguese and spanish cases. In *EMS 2019 Annual Meeting*, volume 16, 2019.

[23] Shashikala L Karunathilake and Harshani RK Nagahamulla. Artificial neural networks for daily electricity demand prediction of sri lanka. In *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 1–6. IEEE, 2017.

[24] Majid Dehghani, Hossein Riahi-Madvar, Farhad Hooshyaripor, Amir Mosavi, Shahaboddin Shamshirband, Edmundas Kazimieras Zavadskas, and Kwok-wing Chau. Prediction of hydropower generation using grey wolf optimization adaptive neuro-fuzzy inference system. *Energies*, 12(2):289, 2019.

[25] Jorida Ajce Konica and Ermira Staka. Forecasting of a hydropower plant energy production with fuzzy logic case for albania. 2017.

[26] B Suprapty, R Malani, and J Minardi. Rainfall prediction using fuzzy inference system for preliminary micro-hydro power plant planning. In *IOP Conference Series: Earth and Environmental Science*, volume 144, page 012005. IOP Publishing, 2018.

[27] Md Anisur Rahman et al. Improvement of rainfall prediction model by using fuzzy logic. *Am. J. Clim. Change*, 9(04):391, 2020.

[28] Raheleh Jafari and Wen Yu. Fuzzy modeling for uncertainty nonlinear systems with fuzzy equations. *Math. Probl. Eng. MATH PROBL ENG*, 2017, 2017.

[29] Namal Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. A novel optimization algorithm: Cascaded adaptive neuro-fuzzy inference system. *Int. J. Fuzzy Syst.*, pages 1–17, 2021.

[30] Cotte Gamage Sarathchandra Gunasekara. Modelling and simulation of temperature variations of bearings in a hydropower generation unit, 2011.

[31] EPN Udayakumara, RP Shrestha, L Samarakoon, and D Schmidt-Vogt. Mitigating soil erosion through farm-level adoption of soil and water conservation measures in samanalawewa watershed, sri lanka. *Acta Agric Scand B Soil Plant Sci.*, 62(3): 273–285, 2012.

[32] EPN Udayakumara, UADP Gunawardena, et al. Reducing siltation and increasing hydropower generation from the rantambe reservoir, sri lanka. *SANDEE*, (113-16), 2016.

[33] Naditha Imbulana, Shakthi Gunawardana, Sangam Shrestha, and Avishek Datta. Projections of extreme precipitation events under climate change scenarios in mahaweli river basin of sri lanka. *Curr. Sci.*, 114(7), 2018.

[34] Anushka Perera and Upaka Rathnayake. Impact of climate variability on hydropower generation in an un-gauged catchment: Erathna run-of-the-river hydropower plant, sri lanka. *Appl. Water Sci.*, 9(3):1–11, 2019.

[35] Bhabishya Khaniya, Isuru Jayanayaka, Pradeep Jayasanka, and Upaka Rathnayake. Rainfall trend analysis in uma oya basin, sri lanka, and future water scarcity problems in perspective of climate variability. *Adv. Meteorol.*, 2019, 2019.

[36] Dashan T Udagedara, Chiaki T Oguchi, and Jagath K Gunatilake. Evaluation of geomechanical and geochemical properties in weathered metamorphic rocks in tropical environment: a case study from samanalawewa hydropower project, sri lanka. *Geosci. J.*, 21(3):441–452, 2017.

[37] K Laksiri, J Gunathilake, and Y Iwao. A case study of the samanalawewa reservoir on the walawe river in an area of karst in sri lanka. In *Sinkholes and the Engineering and Environmental Impacts of Karst*, pages 253–262. 2005.

[38] DD Wijesinghe. Optimization of hydropower potential of samanalawewa project. 2006.

[39] MPW Pathiraja and Wdas Wijayapala. Optimization of the usage of samanalawewa water resource for power generation. In *2016 Electrical Engineering Conference (EECon)*, pages 86–90. IEEE, 2016.

[40] EPN Udayakumara and UADP Gunawardena. Cost–benefit analysis of samanalawewa hydroelectric project in sri lanka: an ex post analysis. *Earth Syst. Environ.*, 2(2):401–412, 2018.

[41] Matthias Jakob Themeßl, Andreas Gobiet, and Armin Leuprecht. Empirical-statistical downscaling and error correction of daily precipitation from regional climate models. *Int J Climatol.*, 31(10):1530–1544, 2011.

[42] IPCC Glossary. Climate change 2014 report fifth assessment report. *IPCC*, 2014.

[43] Jens H Christensen, Fredrik Boberg, Ole B Christensen, and Philippe Lucas-Picher. On the need for bias correction of regional climate change projections of temperature and precipitation. *Geophys. Res. Lett.*, 35(20), 2008.

[44] C Piani, JO Haerter, and E Coppola. Statistical bias correction for daily precipitation in regional climate models over europe. *Theor. Appl.*, 99(1):187–192, 2010.

[45] Claudia Teutschbein and Jan Seibert. Bias correction of regional climate model simulations for hydrological climate-change impact studies: Review and evaluation of different methods. *J. Hydrol.*, 456:12–29, 2012.

[46] Geert Lenderink, Adri Buishand, and W van Deursen. Estimates of future discharges of the river rhine using two scenario methodologies: direct versus delta approach. *Hydrol Earth Syst Sci.*, 11(3):1145–1159, 2007.

[47] Uttam Ghimire, Govindarajalu Srinivasan, and Anshul Agarwal. Assessment of rainfall bias correction techniques for improved hydrological simulation. *Int J Climatol*, 39(4):2386–2399, 2019.

[48] Thomas Lafon, Simon Dadson, Gwen Buys, and Christel Prudhomme. Bias correction of daily precipitation simulated by a regional climate model: a comparison of methods. *Int J Climatol*, 33(6):1367–1381, 2013.

[49] Min Luo, Tie Liu, Fanhao Meng, Yongchao Duan, Amaury Frankl, Anming Bao, and Philippe De Maeyer. Comparing bias correction methods used in downscaling precipitation and temperature from regional climate models: a case study from the kaidu river basin in western china. *Water*, 10(8):1046, 2018.

[50] Rashid Mahmood, Shaofeng Jia, Nitin Kumar Tripathi, and Sangam Shrestha. Precipitation extended linear scaling method for correcting gcm precipitation and its evaluation and implication in the transboundary jhelum river basin. *Atmosphere*, 9(5):160, 2018.

Chapter 5

Mahaweli River flood prediction, a case study in Sri Lanka

Chapter 5 focuses on the application of rainfall-runoff predictions. This study mainly focused on a case study on the Mahaweli River in Sri Lanka. This work is currently (update 2022/12/21) under review in PLOS ONE journal.

5.1 Introduction

Natural disasters often occur due to recent climate changes. Several studies have focused on climate change and its' effect detection where Remote sensing methods are highly used in these methodologies [1].

Floods are frequently observed in natural disasters. However, they are one of the direct outcomes of the rainfall-runoff (R-R) process [2]. Due to their severity and frequent occurrence, flood prediction has taken significant attention in R-R modelling [3]. Even though they are natural disasters, their severity has been impacted by anthropogenic activities. Flow hydrographs are drastically changed to have higher peaks quickly due to ongoing urbanization [4–6]. Flash floods are often in urbanized areas [7, 8]. Hence, urbanization is one of the most impacting factors in today's floods.

In addition to urbanization, changing climate has adversely impacted today's floods. Some regions receive higher and intensified rainfall events [9–12] whereas some other areas receive reduced rainfall events[13, 14] due to ongoing climate change. Frequent floods are expected in areas with projected increased rainfall events. Many studies in the literature support this observation [15–17].

Therefore, accurate modelling of runoff-rainfall relationships to catchments is in high demand. It is important to note that each catchment has to be modelled to find its R-R relationship. Commercial and non-commercial hydrological computer packages are available to simulate the R-R relationships of catchments. However, these computer packages require various data related to digital elevation models, soil data, meteorological data, and discharge data. [18]. The accuracy of the catchment models is highly varied due to the quality of catchment data [19]. Only some catchments are gauged to have meteorological and discharge data and other catchment characteristics on a temporal and spatial basis. Thus, the catchment models always need help achieving the required accuracy to model the runoff and then predict the floods.

In the event of limited data, soft computing [20, 21], and machine learning techniques [22–25] are helpful to model the R-R processes. R-R processes can be modelled only using the known rainfall and measured discharges and, importantly, without any catchment characteristics. Hence, numerous methodologies under soft computing and machine learning have been developed using various algorithms and study cases. One of these data-driven methods is the artificial neural network (ANN), which has been used in various fields, including hydrology and water resources. It has gained popularity because it can address, model, and forecast stochastic and nonlinear situations in the system [26–32] The algorithm does not replace conceptual watershed modelling of the impossibility of describing the catchment's internal structure and handling the data disseminated relating to the physical properties. Nevertheless, they have gained acceptance as a practical substitute for conceptual models for forecasting because of numerous benefits, such as the ability to produce simple and accurate models [33] and the computation speed [34]. Additionally, this study has demonstrated its strength and ability to mimic hydrological events. As a result, ANN models are suggested for rainfall-runoff modelling due to their straightforward designs and accuracy, enabling addressing the issues of managing water resources.

In order to create ANN models, most studies have used feed-forward and backpropagation (FFBP) networks. Although relatively well known for their ability to anticipate floods, neither model's performance in a particular application has been determined [28]. Since several learning methods may be used to improve ANN, there is still a wide range of probability. Gradient descent (GD) is frequently used in neural network training at the backpropagation stage [35]. GD has been used in recent years to increase the potential of the backpropagation algorithm. However, the GD may experience problems with convergence, training technique slowdown, overfitting, and stocking inside local minima. The performance of the training algorithm can lower the performance when the structure of the model is complex, and the parameter set is significant [36–38].

Moreover, Feed-forward deep neural networks (FF-DNNs) have been used widely in climate change-related studies. A case study in Kastoria Lake in Greece used FF-DNN to predict dissolved oxygen. They have obtained maximum NSE efficiency of 0.89 [39]. Forecasting of dissolved oxygen was studied using three methods such as the Autoregressive integrated moving average (ARIMA) method, Transfer Function (TF) method, and NN method [40]. They concluded that the ARIMA method provides significant results compared to TF and NN. Additionally, A combination of tools such as remote sensing, weather forecasting, and Artificial Intelligence was used to improve irrigation management in Mediterranean Basins. This study suggests that comprehensively using these tools can enhance the irrigation system rapidly [41].

Recently, several novel evaluations of CNN models were implemented: the Extreme Gradient Boosting (XGBoost) and CNN-transformer. These algorithms have been widely tested for uncertain and nonlinear data. Many studies recommended ANFIS as a highly accurate algorithm for predictions [20, 21]. Xuan-Nam et al. (2010) [42] have proposed an ML model for blast-induced ground vibration predictions in quarries. They have employed several state-of-the-art algorithms, such as Moth-flame optimization-based ANFIS, XGBoost, ANN, and SVM. The study showcased that the ANIFS-based algorithm outperformed the other model with an accuracy of 98.62%. Moreover, two environmental types of research have been introduced by Hamid et al. (2020)[20] and Junliang et al.(2019)[21] employing ANFIS and XGBoost algorithms.

On the other hand, Genetic Algorithms (GA) in the hydrological sciences have been the subject of several investigations to train (ANN) rainfall-runoff models that are more accurate than backpropagation technique-based ANN models in anticipating the quotidian flow [43] using natural code GAs. In conjunction with intelligence approaches, the GA has developed into a potent tool for modelling and optimizing complicated processes [44–46]. It is commonly used in ANN to enhance efficiency by tuning the parameters [47, 48]. Roy and Singh [36] developed a novel hybrid metaheuristic method for simulating the rainfall-runoff process that integrates Biogeography-Based Optimization (BBO), Particle Swarm Optimization (PSO), and grey wolf optimizer (GWO) combining ANN and Adaptive Network-based Fuzzy Inference Systems (ANFIS). Moreover, three optimization algorithms integrated with ANFIS were introduced for rainfall-runoff predictions, namely, Differential Evolution algorithm based ANFIS (ANFIS-DE), Particle Swarm Optimization based ANFIS (ANFIS-PSO), and Genetic Algorithm based ANFIS (ANFIS-GA) [49]. Investigating and contrasting these models in hydrology is strongly advised because the different algorithms have various advantages and distinct methods for complex modelling phenomena. The investigations in hydrology, particularly rainfall-runoff modelling, are still in the early stages. Hence, the computational analysis has to be comprehensively conducted for a better outcome. Therefore, this research study aims to contribute to scientific society by achieving the following objectives.

1. Designing and developing an accurate, low computational complex machine learning model for rainfall-runoff forecasting. (The Cascaded-ANFIS)

2. Conducting comprehensive experiments to support the proposed algorithm using three regression algorithms (Long Short-Term Memory (LSTM), Grated Recurrent Unit (GRU), and Recurrent Neural Networks (RNN)) using an important river basin in Sri Lanka

3. Predict the future water levels for the near-future (2022 - 2030) and mid-future (2031 - 2050) using Shared Socio-economic pathways (SSP245 and SSP585) and then analyze the flood events in the future.

5.2 Methodology

5.2.1 Problem formulations

The following relationship shown in Equation 5.1 was modelled using the Cascaded-ANFIS algorithm. The relationship was trained using the ground-measured rainfall and water level. Subscript t in Equation 5.1 denotes the time domain of the R-R relationship.

$$WaterLevel_t = f(RainFall_{i,t}) \tag{5.1}$$

However, it is well noted that time domains can be shifted from rainfall to runoff from that rainfall due to the catchment characteristics like river length, catchment area, land use patterns, and soil type. The travel time of a particular rainfall event has to be clearly understood.

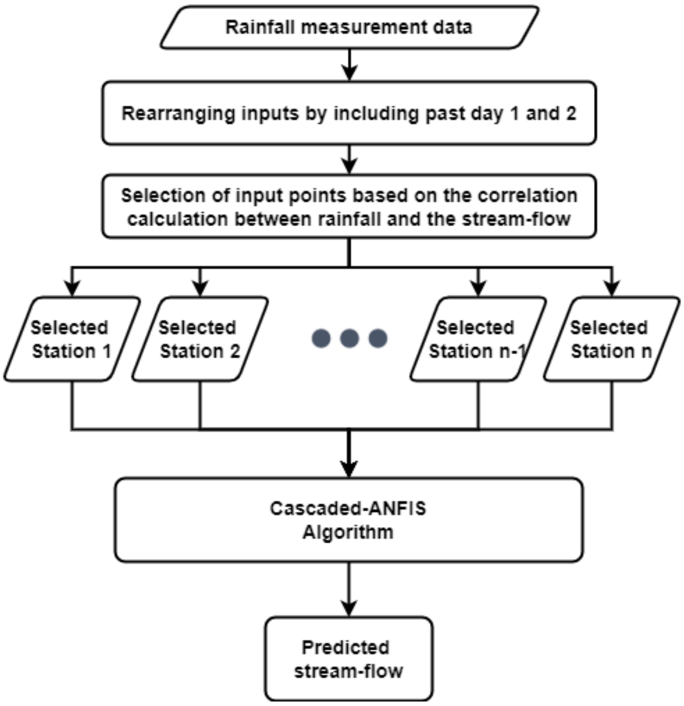


Figure 5.1: The overall structure of the Cascaded-ANFIS implementation using the selected inputs.

Figure 5.1 presents the flowchart for the developed Cascaded-ANFIS model. As shown in the Figure, the rainfall data is used as the primary input of the system. Then the input data are re-arranged with a delay of one day and two days. The inputs were then removed based on the computation of the correlation between each input and the output of the flow level. A minimal correlation of 0.40 between an input and an output was used in this case. The selection methodology of inputs is discussed in later sections.

5.2.2 Comparative analysis to identify the best algorithm

Three regression algorithms (Long Short-Term Memory (LSTM), Grated Recurrent Unit (GRU), and Recurrent Neural Network (RNN)) together with the Cascaded-ANFIS algorithm were used to formulate the R-R relationship. These ML algorithms were considered in this study due to a few specific reasons, such as algorithms being similar and easy implementation. Moreover, they are low in weight and can be processed in a general computer without GPU support. Table 10.1 shows that the same parameter values were considered for LSTM, GRU, and RNN tuning. These parameters were selected based on trial and error methods. Each parameter is tested with the datasets used in this study and employs the optimum value.

The Cascaded-ANFIS used three gaussian membership functions for each input in the system. The whole cascades were ten to achieve satisfactory accuracy and error value.

Table 5.1: Parameter settings of the algorithms used in this study.

Algorithm	Parameters	
RNN/LSTM/GRU	Optimizer	adam
	Learning rate	0.0001
	Activation	relu
	Batch size	72
	Epoches	1000
Cascaded-ANFIS	Iterations	100
	Membership Functions	Gausian
	Number of Membership Functions	3
	Number of Cascades	10
	Step Size	0.1
	Decrease rate	0.9
	Increase rate	1.1

5.2.3 Mahaweli River sub-catchment analysis

Localized floods can be observed in sub-catchments in Figure 5.3a and 5.3b without showcasing major floods downstream of the river due to the catchment characteristics. Therefore, the downstream river gauge may not observe any flood situation. However, upstream sub-catchments might have experienced localized floods. Therefore, it is essential to cluster larger catchments into sub-catchments and then analyze them separately. This scenario was analyzed in this research work and formulated Equation 5.1 for sub-catchments.

5.2.4 Flood identification

According to the desinventar dataset of natural disasters [50], there has been significant damage due to flooding in Sri Lanka. In most cases, the damage has increased due to unexpected heavy rainfall and poor irrigation management. The database reveals that in the past events from 2005 to 2018, there was at least one death due to flooding. The highest number of deaths, injured and missing personals were recorded in 2017, with 67, 73, and 63, respectively.

Historical water levels were analyzed to define threshold water levels to identify floods in the basin. Here, water levels were considered because the authorities recorded the data as water levels instead of the water flows. If the water levels or stream flows exceed the threshold, that flow may be a flood. However, this can be confirmed with the ground-measured discharge data and by comparing flood data to the catchment. Nevertheless, many countries do not have these flood databases, so there can be some issues with the accuracy [51].

5.2.5 Shared Socio-economic Pathways (SSP) Climate Data Extraction

IPCC's sixth report [52] presented a new set of scenarios based on greenhouse gas emissions to project the future climates until 2100. Practitioners who engage with future climate data may investigate climate changes across a range of quite diverse futures thanks to the availability of climate forecasts for numerous Shared Socio-Economic Pathways (SSPs). These SSPs are titled SSP1, SSP2, SSP3, SSP4, and SSP5 under several Socioeconomic Pathways. SSPs describe potential future growth pathways for human cultures. A set of models combine assumptions on the ambitions for reducing the impact of climate change with predictions about how population, education, energy usage, technology, and other factors may evolve over the next century. Various conceivable future climates, from a pessimistic high-carbon scenario to a low-carbon one that satisfies the goals of the 2015 Paris Agreement, are described in the climate change forecasts from these scenarios [53, 54].

The Representative Concentration Pathways, or RCPs, or earlier projections of greenhouse gas concentration, are improved upon by SSP-based scenarios. To investigate the consequences of various emission trajectories or emissions concentrations, RCPs were explicitly created for the community of climate modellers. It is challenging to relate social trends such as population growth, educational attainment, and government policies to climate objectives like limiting global warming to below 2 $°C$ since the socioeconomic factors used to establish RCPs need to be standardized. To address this, SSPs outline how societal decisions might alter Radioactive Forcing towards the end of the century. As a result, SSPs were built on RCPs to enable a uniform comparison of societal decisions and the degrees of climate change they cause. These SSP data are used in various recent research studies such as flood forecasting [55], land use optimization [56], and prediction of air pollution for the future [57]. Climate change research [57]. According to these studies, the reliability of SSP data is much higher than the RCP data. Therefore, this study employed SSP projections for daily rainfall data acquisition [58, 59]. Here, two

SSP scenarios have been used for the data acquisition, such as SSP2-4.5 and SSP5-8.5. SSP2-4.5 represents the low carbon impact globally, while SSP5-8.5 is the high carbon scenario.

5.2.6 Bias Correction

The extracted rainfall data under SSP2-4.5 and SSP5-8.5 were corrected using linear bias correction factors. Usually, the data extracted from climate models may have some systematic errors [60]. Therefore, the model's extracted climate data are corrected for bias using the ground-measured climate data. Various bias correction techniques are available [61]; however, the linear bias correction method was selected in this research work. Equation 5.2 gives the simple mathematical formulation for linear bias correction. More details on this can be found in Chaturanika et al. [62].

$$RF^{*}_{sim}(d) = RF_{sim} \times \frac{\mu_m(RF_{obs}d))}{\mu_m(RF_{his}(d))} \tag{5.2}$$

Where RF, d, μ_m, his, obs, and sim are rainfall, daily, long-term monthly mean, raw SSP data, observed/measured data, and raw RCM forecast. The symbol $*$ denotes the bias-corrected datasets.

5.2.7 Projected water levels and floods

Bias-corrected SSP rainfall data were fed to the developed R-R relationship in Equation 5.1. Based on these future rainfalls under two SSP scenarios, the stream flows in the means of water levels were predicted for future years. These predicted water levels for the whole catchment were tested for the extreme values in the time series and then identified localised and downstream floods. These predicted floods are given for the near future (from 2022-2030) and mid-future (2031-2050).

5.3 Case Study

Sri Lanka is a country blessed with water resources. However, heavy monsoon rainfall drives many rivers into floods, and annual floods are quite often [63]. Sri Lanka has many rivers, tanks and lakes, and these watersheds are flooded during the monsoon periods. Several deaths and excessive structural damage are annually reported due to extreme weather conditions. Sri Lanka has 103 rivers, and the total length of the rivers is around

4500 km. The longest river in Sri Lanka is the Mahaweli River. It is 335 km long and covers a 10488 km^2 river basin which covers almost one-fifth of the total area of the island [64, 65]. The river has several branches along the way to the sea. 40% of the total electricity demand of Sri Lanka is provided by the hydropower generated by the Mahaweli River. Nevertheless, the Mahaweli River is known to provide a vast water supply for the cultivation of crops such as rice and vegetables [66]. Several Mahaweli River developments have been for hydroelectric generation and irrigation purposes. Many dams were constructed along the river to enhance energy generation, which led to flood risk changes. Kothmale dam was one of those constructed to generate electricity; however, indirectly, it has mitigated the floods downstream [67]. The Mahaweli River was selected for this research study due to its importance in many utilities and its frequent floods in the northeastern monsoon period (from December to February).

5.3.1 Study area and sub-catchments

The Mahaweli River starts from the central hills of Sri Lanka with several small creeks. Agra Oya from Horton Plains is one of the starting creeks of the Mahaweli River. The river reaches the Bay of Bengal on the southwestern side of Trincomalee Bay. The bay includes the first of several submarine canyons, making Trincomalee one of the finest deep-sea harbours in the world. As part of the Mahaweli Development program, the river and its tributaries are dammed at several locations to allow irrigation in the dry zone, with almost 1,000 km^2 (386 sq mi) of land irrigated. Figure 5.3 presents the primary catchment and sub-catchments, whereas Figure 5.2 shows the catchment of the Mahaweli River basin.

Two sub-catchments were identified along two tributaries of the Mahaweli River. The catchment above Peradeniya (for Kothmala Oya and other parts upstream creeks of Mahaweli River) is given in Figure 5.3a while the catchment above Thaldena for Badulu Oya is given in Figure 5.3b. The sub-catchment at Peradeniya is in the wet zone of the country; thus, heavy rainfall can be experienced. However, the sub-catchment at Thaldena is in the wet and intermediate zone. Thus, the rainfall in that sub-catchment is not as high as that at Peradeniya. However, these two sub-catchments are essential in the aspects of terrain, land use, and urbanization. In addition, two flow gauges can also be found in these two sub-catchments.

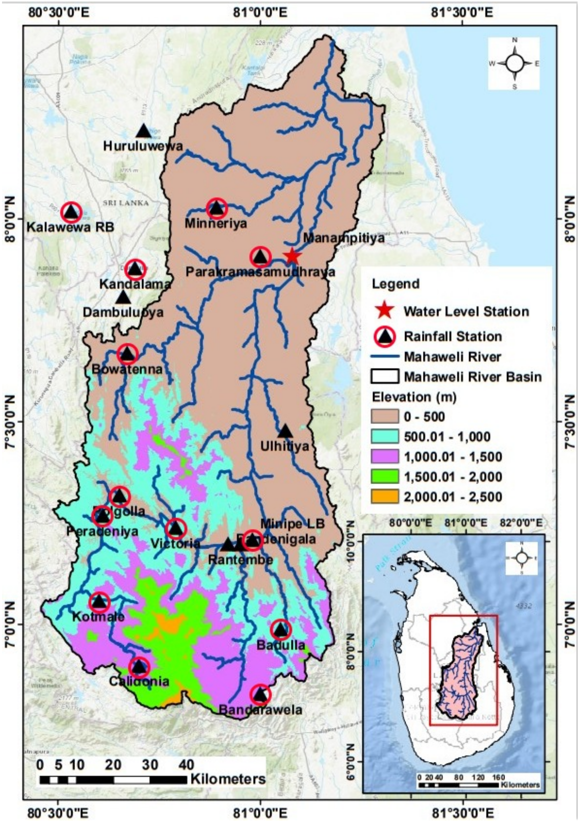
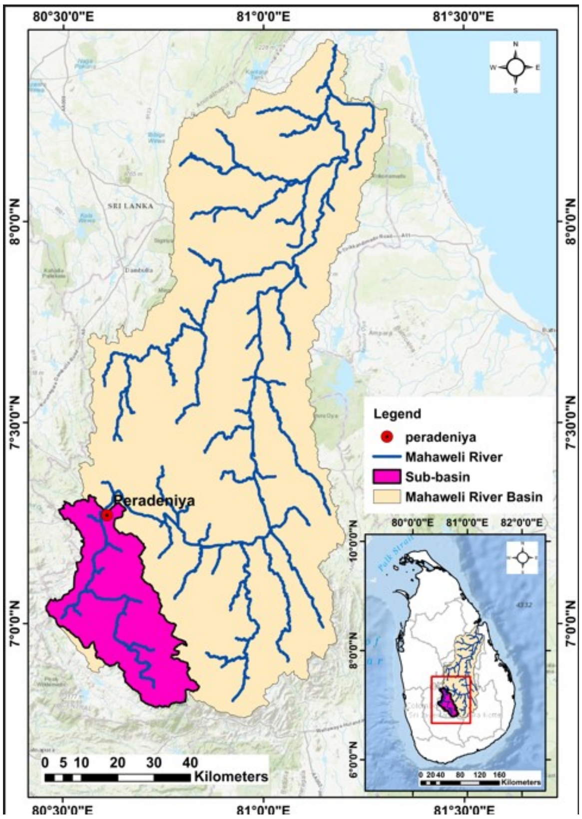


Figure 5.2: Study area of the Mahaweli river catchment map

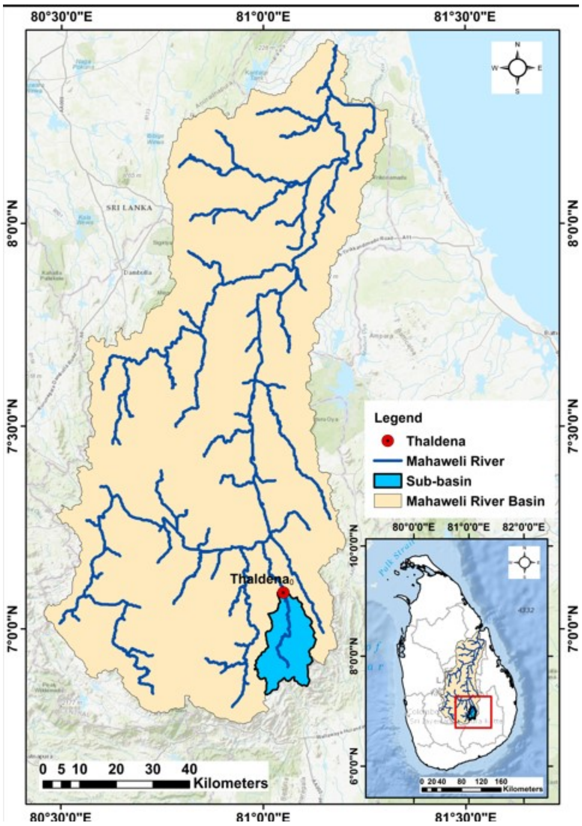
5.3.2 Data

Figure 5.2 shows rain gauges for the Mahaweli River basin. Due to the unavailability of complete data in most of the years, the daily rainfall data from 2000 to 2017 were purchased from the Department of Meteorology, Sri Lanka. The missing data percentage for the selected years was less than 1%. The rain gauges were selected to represent the whole catchment covering as much as its area. In addition, the stream flow gauge at Manampitiya was selected to model the R-R relationship. This is the most downstream stream flow gauge available. The water levels at the station were purchased from the Department of Irrigation, Sri Lanka. Furthermore, two water level measuring stations were identified for the selected two sub-catchments: Pereadeniya and Thaldena (refer to Figures 5.3a and 5.3b). The water levels for these two stations were also purchased for the same period from the Department of Irrigation, Sri Lanka.

A descriptive analysis of the dataset used in this analysis is shown in Table 5.2. There were 6207 data samples in the dataset. The water levels are presented in centimetres, whereas the rainfalls are presented in millimetres. Moreover, several homogeneity tests were conducted, such as the Standard normal homogeneity test (SNHT), Buishand range



(a)



(b)

Figure 5.3: Sub Catchment Study areas; (a) Catchment map at the Peradeniya sub-catchment; (b) Catchment map at the Thaldena sub catchment

Table 5.2: Descriptive analysis of the data for the Mahaweli River basin

Variable	Sample Data	mean	std	min	25%	50%	75%	max
Peradeniya	6207	5.27	13.56	0	0	0	3.9	194.3
Minneriya	6207	4.28	14.09	0	0	0	0	210
Calidonia	6207	5.84	11.87	0	0	0	7	144.6
Parakrama samudhraya	6207	5.13	15.69	0	0	0	0	222
Kandalama	6207	4.55	13.78	0	0	0	0	198.6
Kalawewa RB	6207	3.52	11.51	0	0	0	0	166
Bowatenna	6207	5.09	15.29	0	0	0	1.1	242.5
Kotmale	6207	6.50	13.95	0	0	0	7	191
Polgolla	6207	4.61	11.98	0	0	0	3	170
Randenigala	6207	4.61	13.70	0	0	0	1.3	270.9
Victoria	6207	3.98	11.89	0	0	0	1.3	295
Badulla	6207	5.05	11.94	0	0	0	3.7	195.9
Bandarawela	6207	4.48	10.36	0	0	0	3.4	134.9
Mannampitiya Water Level	6207	0.92	1.94	0.02	0.17	0.34	0.80	26.93

(BR) test, Pettitt test, and von Neumann ratio (VNR) test to evaluate the dataset before employing it in training models.

Due to the missing data in a significant time frame, few rainfall stations were omitted in the evaluation of the case study. The missing data were presented in Huruluwewa, Dambuluoya, Ulhitiya, Minipe LB, and Rantembe. Therefore, as shown in Table 5.2, 13 rainfall station data were considered as the inputs.

The correlation calculation in subsection 5.2.3 is given in Table5.3. The selected outputs are highlighted with a minimum of 0.4 correlation. Twelve inputs were selected using the correlation method to train the R-R model. The trial and error method made the selection based on the correlation. At a correlation value of 0.40, the maximum accuracy was obtained. Then the general structure of the Cascaded-ANFIS was used to generate the final outputs of predicted water levels. Additionally, according to the literature, it is considered negligible if a correlation is 0.30 or below. Therefore, 0.40 and above values were considered safe marginal inputs in the system [68].

5.3.3 Recent floods for the river basin

Figure 5.4 shows the annual water level measurements at each of the observation points, such as the primary catchment of Mahaweli River (Mannampitiya) and sub-catchments of Mahaweli River (Peradeniya and Thaldena). It can be seen that Mannampitiya water outlets record a higher level of water when compared with the sub-catchments. As indicated by the figure (refer to the rectangular section), the water levels on some of

Table 5.3: Correlations between inputs and the flow levels

RF Stations	Correlation		
	t	t-1	t-2
Peradeniya	0.14	0.25	0.22
Minneriya	0.30	0.41	0.38
Calidonia	0.08	0.16	0.14
Parakramasamudhraya	0.34	0.49	0.43
Kandalama	0.28	0.40	0.36
Kalawewa	0.22	0.31	0.27
Bowatenna	0.32	0.47	0.40
Kotmale	0.08	0.15	0.12
Polgolla	0.19	0.33	0.30
Randenigala	0.29	0.48	0.48
Victoria	0.29	0.49	0.45
Badulla	0.25	0.42	0.42
Bandarawela	0.17	0.29	0.28

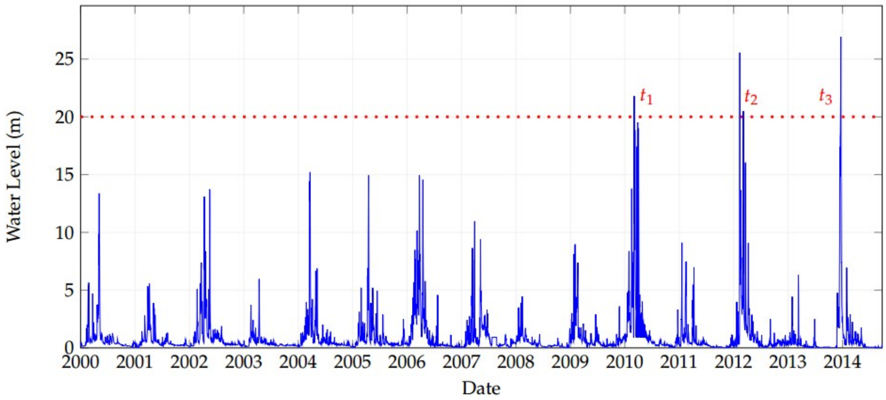
the days of 2011 (21.8 m on 10/01/2011), 2012 (25.6 m on 18/12/2012 and 21.7 m on 27/12/2012), and 2014 (26.9 m on 27/12/2014) were higher than 20m. These can be identified as flood thresholds to the Manampitiya river gauge.

Sub-catchments Pereadeniya and Thaldena showcased some higher water levels comparable to the higher water levels at Manampitiya; however, some differences can also be observed (refer to Table 5.4). Thaldena has not showed a significantly higher water level in 2012, but higher water levels were observed at Manampitiya during the same time (t_1, t_2, and t_3 in Figure 5.4a). Similar trends can be observed in Peradeniya too. Therefore, the analysis of sub-catchments for floods is highly justified. Comparable observations have led the authors to define flood thresholds for Peradeniya and Thaldena. The threshold for Peradeniya was considered 6 m, while 3 m was considered for Thaldena. The flood events were identified in Peradeniya and presented as t_1, t_2, and t_3 in Figure 5.4b (6.7 m on 03/06/2013, 6.9 m on 14/09/2013, and 6.7 m on 26/12/2014) while two incidents were identified for Thaldena and presented as t_1 and t_2 in Figure 5.4c(3.1 m on 02/02/2011 and 3.5 m on 26/12/2014).

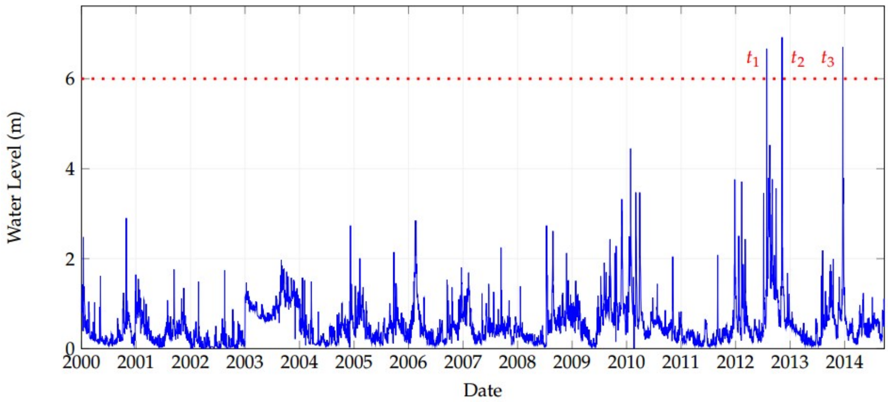
5.4 Experimental Results

5.4.1 Evaluation Parameters

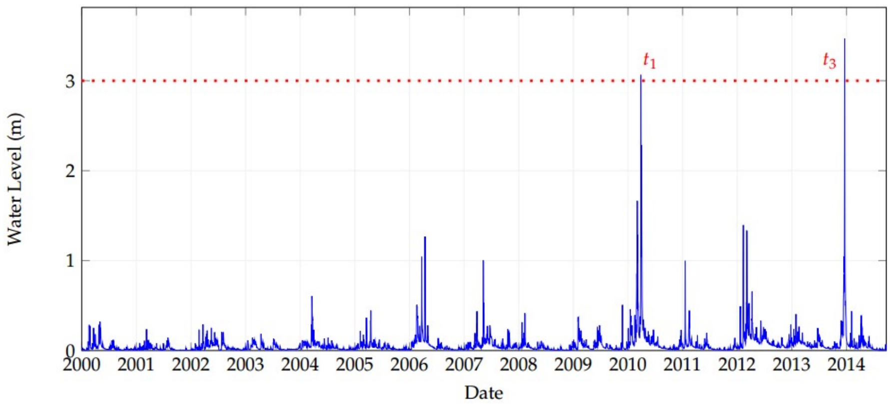
The algorithm performances were then tested by several metrics including root mean square error (RMSE), bias, Nash-Sutcliffe efficiency (NSE), Kling- Gupta Efficiency



(a) for Mannampitiya



(b) for Peradeniya



(c) for Thaldena

Figure 5.4: Historical Water Level measurements from year 2000 to 2015: (a) at the Manampitiya water level measurement station, (b) at the Peradeniya water level measurement station, and (c) at the Thaldena water level measurement station

Table 5.4: Water levels at river gauges

Date	Water Levels (m)		
	At Manampitiya	At Peradeniya	At Thaldena
10/01/2011	**21.8**	2.5	0.98
02/02/2011	19.2	1.6	**3.1**
18/12/2012	**25.6**	3.7	1.4
03/06/2013	0.1	**6.7**	0.1
14/09/2013	0.4	**6.9**	0.05
26/12/2014	**21.7**	**6.7**	**3.5**
27/12/2014	**26.9**	3.3	1.2

Bold values are identified as floods

(KGE), and correlation coefficient (R). These performance evaluating metrics are given in Equations (7.6), (7.2), (7.3), and (7.1).

$$RMSE = \sqrt{\frac{1}{q}\sum_{t=1}^{q}(u(t) - \bar{u}(t))^2} \tag{5.3}$$

$$bias = \frac{\sum_{j=1}^{k} u(t) - \bar{u}(t)}{\sum_{j=1}^{k} u(t)} \tag{5.4}$$

$$NSE = 1 - \frac{\sum_{j=1}^{k}(u(t) - \bar{u}(t))^2}{\sum_{j=1}^{k}(u(t) - \bar{v}(t))^2} \tag{5.5}$$

$$R = \frac{\sum(v(t) - \bar{v}(t))(u(t) - \bar{u}(t))}{\sqrt{\sum(v(t) - \bar{v}(t))^2 \sum(u(t) - \bar{u}(t))^2}} \tag{5.6}$$

Where $u(t)$ is the predicted parameter, $\bar{u}(t)$ is the mean of predicted parameter$v(t)$ is the measured parameter, k is the population size, and $\bar{v}(t)$ is the mean of the measured parameter. The correlation coefficient (R) represents the goodness of fit. It varies from -1 to 1; the best is when it becomes 1. Bias tells the differences between predicted to measured values. The ideal bias value is 0, and 1 becomes the worst. NSE calculates the perfectness of the match between actual and prediction. The results of the NSE can vary between minus infinity being the worst and 1 being the ideal [69]. KGE is a combined calculation of three primary parameters: NSE, bias, and coefficient of variation. Recently it has been used rapidly in hydrological model performance calculations [70].

5.4.2 Performance Evaluation

5.4.2.1 Correlation of Coefficients calculation for the main catchment

The primary catchment of the Mahaweli River consists of 13 rain gauges, all of which were used to predict the water level at Manampitiya. As mentioned in the previous sections, the experiment was designed to identify the best R-R prediction algorithm. The algorithms used in this study are LSTM, GRU, RNN, and Cascaded-ANFIS. Figure 5.5 shows the coefficient of correlation of the predicted water to the ground-measured water level at the Manampitiya river gauge.
Figure 5.6 presents the prediction accuracy under combined scenarios which were initially identified as per Table 5.3 for the predicted water levels at Manampitiya.

5.4.2.2 Correlation of Coefficients calculation for sub-catchments

Figure 5.7 and 5.8 shows the prediction accuracy of water levels for each algorithm for the sub-catchments Peradeniya and Thaldena.
Additionally, a few other parameters were used for the evaluations of the results such as Bias, NSE, RMSE, and KGE. The evaluation results are presented in Table 5.5.

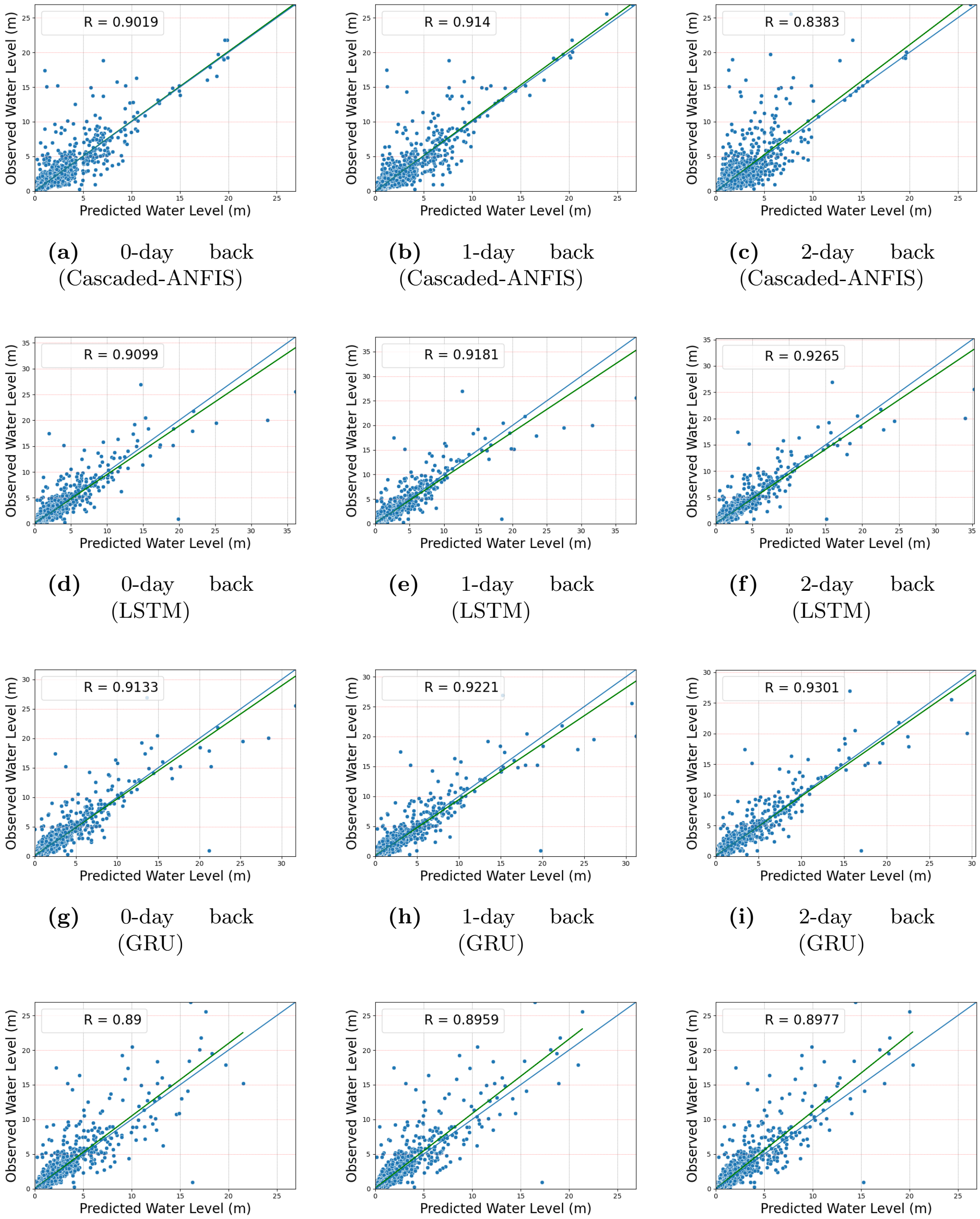
5.4.3 Projected water levels at Manampitiya

Figures 5.9 illustrate the projected future water levels at Manampitiya under the two SSP scenarios for the near future (2022-2030) and mid-future (2031-2050). These results project some exciting interpretations. None of the scenarios presents extreme flood situations for any year from 2022 to 2050. This is very surprising. This can be due to several reasons, including the future data quality and bias correction technique. However, these strange results imply that the researchers conducted some extensive projected flood analysis based on the ground-measured flow situations. In addition, the R-R model can be implemented for Representative Concentration Pathways (RCPs) and then analyze the differences.

5.5 Discussion

5.5.1 Model Evaluations

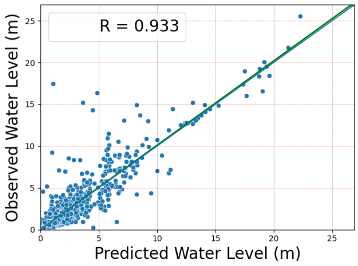
According to Figure 5.5i, it can be seen herein that the best prediction was performed by the GRU algorithm with an R of 0.9301. In addition, the LSTM algorithm with 2-day



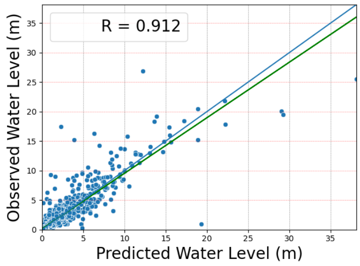
(a) 0-day back (Cascaded-ANFIS)
(b) 1-day back (Cascaded-ANFIS)
(c) 2-day back (Cascaded-ANFIS)
(d) 0-day back (LSTM)
(e) 1-day back (LSTM)
(f) 2-day back (LSTM)
(g) 0-day back (GRU)
(h) 1-day back (GRU)
(i) 2-day back (GRU)
(j) 0-day back (RNN)
(k) 1-day back (RNN)
(l) 2-day back (RNN)

Figure 5.5: Mahaweli catchment water level prediction and observed values with calculated correlation coefficient (R): 0-days (current day inputs), 1-day (the current day and past 1-day inputs), and 2-day (the current day and past 1 and 2-day inputs)

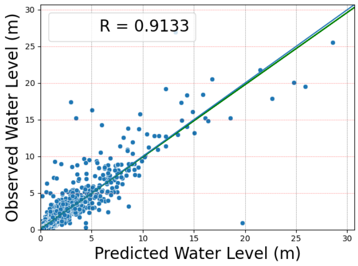
back rainfall data (t-2 scenario) performed as the second best with 0.9265 (refer to Figure 5.5l). Interestingly, as per Figure 5.5b, the Cascaded-ANFIS algorithm showcased its highest R-value at 0.9140 for 1-day back rainfall data (t-1 scenario). However, it can



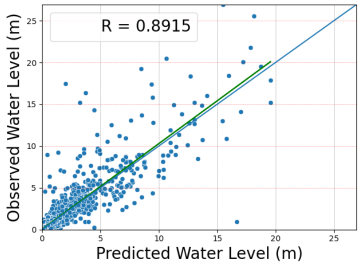
(a) Cascaded-ANFIS test predictions for selected inputs



(b) LSTM test predictions for selected inputs

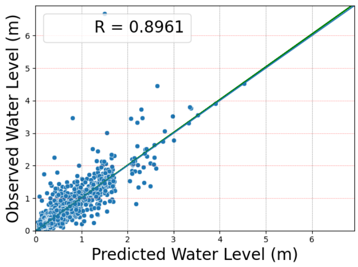


(c) GRU test predictions for selected inputs

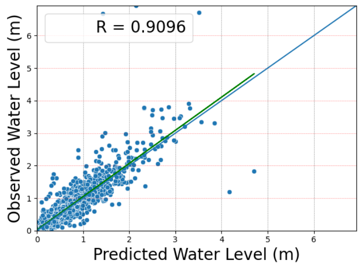


(d) RNN test predictions for selected inputs

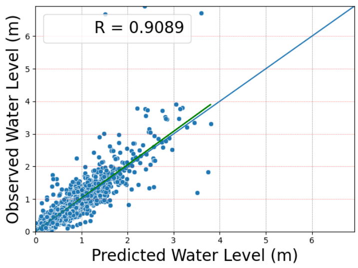
Figure 5.6: Prediction accuracy for water levels combined scenarios at Manampitiya: (a) Cascaded-ANFIS; (b) LSTM; (c) GRU; (d) RNN



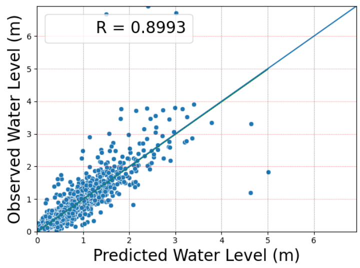
(a)



(b)



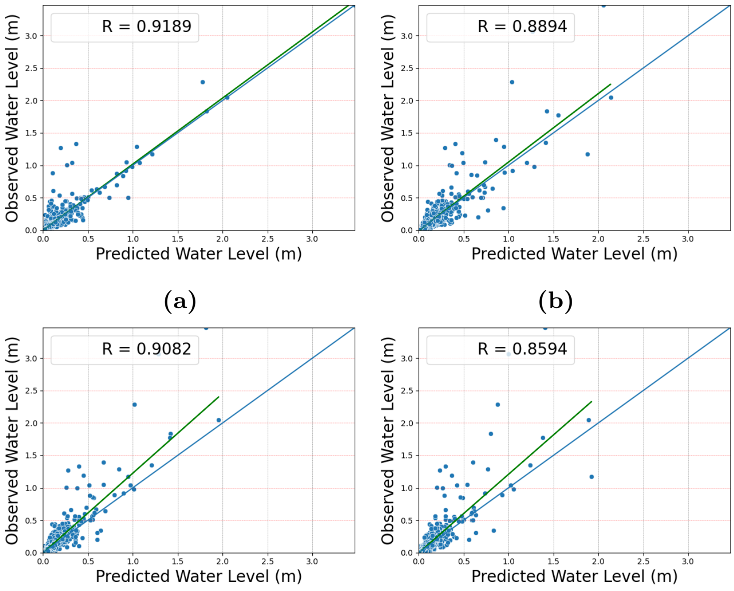
(c)



(d)

Figure 5.7: Prediction accuracy for water levels at Peradeniya: (a) Cascaded-ANFIS; (b) LSTM; (c) GRU; (d) RNN

be clearly understood that three scenarios separately cannot be used to model the R-R relationship. In other words, the rainfall which occurs two days back for the most upstream location can reach Manampitiya on the current day. Similarly, rainfall received



(a) (b)

(c) (d)

Figure 5.8: Prediction accuracy for water levels at Thaldena: (a) Cascaded-ANFIS; (b) LSTM; (c) GRU; (d) RNN

one day back in another location can reach Manampitiya on the current day. Therefore, a combination of these three scenarios has to be considered.

As in the selected rainfall gauge analysis, it was clear that the results were more consistent and accurate. The Cascaded-ANFIS algorithm-based prediction model had an R of 0.933 for selected inputs (refer to Figure 5.6a). GRU, LSTM, and RNN showed R values of 0.9133, 0.9120, and 0.8915, respectively, and were outperformed by Cascaded-ANFIS. Therefore, the Cascaded-ANFIS algorithm can be used effectively to predictions of water levels.

The sub-catchment correlation coefficient analysis in Figures 5.7 and 5.8 shows that the Cascaded-ANFIS algorithm has outperformed the other three algorithms in predicting water levels at the sub-catchment level. In Figure 5.8, the correlation coefficients were found to be 0.9188 for Cascaded-ANFIS, 0.8894 for LSTM, 0.9082 for GRU, and 0.8594 for RNN. Therefore, the water level prediction for the Thaldena sub-catchment also succeeded by the prediction model developed based on the Cascaded-ANFIS algorithm. The proposed algorithm shows the least RMSE with 0.66. The proposed algorithm also scored the highest NSE and KGE values, with 0.87 and 0.90. The second-best performances were shown by the GRU algorithm having RMSE, NSE, and KGE as 0.79, 0.83, and 00.88. When considering the bias factor of the predicted outputs, the Cascaded-ANFIS model shows a significantly low value of 1.52. This low score for the bias provides a certification that the model can predict the water levels with higher accuracy and lower bias. The overall results are shown in Table 5.5.

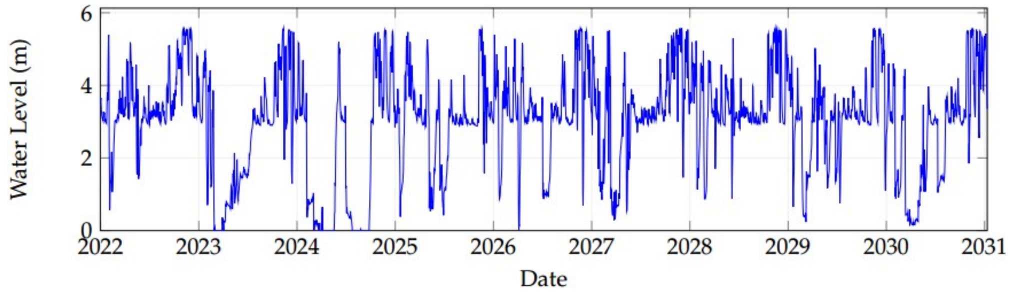
Table 5.5: The evaluation results of the study; Percent bias value (Bias), Root Mean Squared Error (RMSE), Nash-Sutcliffe efficiency (NSE), Kling-Gupta efficiency (KGE), Correlation Coefficient (R)

Algorithm	Bias	RMSE	NSE	KGE	R
Cascade-ANFIS	1.52	0.66	0.87	0.90	0.93
Grated Recurrent Unit	4.09	0.79	0.83	0.88	0.91
Long Short Term Memory	8.88	0.81	0.83	0.87	0.91
Recurrent Neural Networks	6.66	0.88	0.79	0.82	0.89

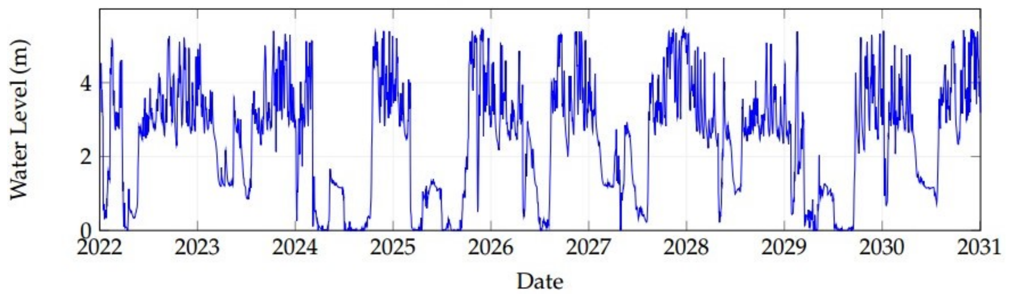
5.5.2 Forecasting of the river water level

Let the predictions be accurate (assumed). Then, there is a severe issue in the water levels, thus the river flow at Manampitiya. The average water levels for Manampitiya are around 10 m (from its historical data). However, the projected water levels are around 6 m (60% of the average). Therefore, drought conditions can be projected. The predicted outcomes of the trained model can be a result of the dataset. The dataset provides a short range of rainfall data. Therefore, more than the sample size may be needed to train a perfect R-R model. However, this cannot be considered a conclusion of this study. Even though the prediction accuracy is good in the Cascaded-ANFIS model, future data quality is critical in a solid prediction. Therefore, Figures 5.9 cannot be treated as a conclusion of this study.

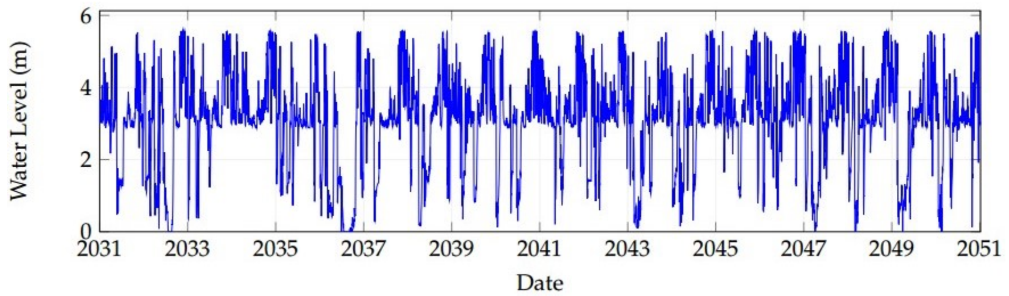
However, these water levels were presented in Figure 5.10 shows the forecasting of water levels at Manampitiya for the projected rainfalls. From the year 2031 to 2050, forecasting is shown in Figures 5.10c and 5.10d respectively for SSP2-4.5 and SSP5-8.5. The X-axis contains 365 ticks representing days of the year, and the scale bar on the right side of Figure 5.10 showcases the intensity of the water level. During the northeaster monsoon (December to February), the water levels can be observed at higher levels, as predicted at Manampitiya. However, the SSP5-8.5 scenario has projected lower water levels for mid-year, reaching less than 1 m. These can be droughts. However, the SSP5-8.5 is a higher scenario for fossil-fueled development. This observation cannot be seen in the SSP2-4.5 scenario. The key observations are indicated using black and white squares where black being lower water level periods and white being higher water level periods.



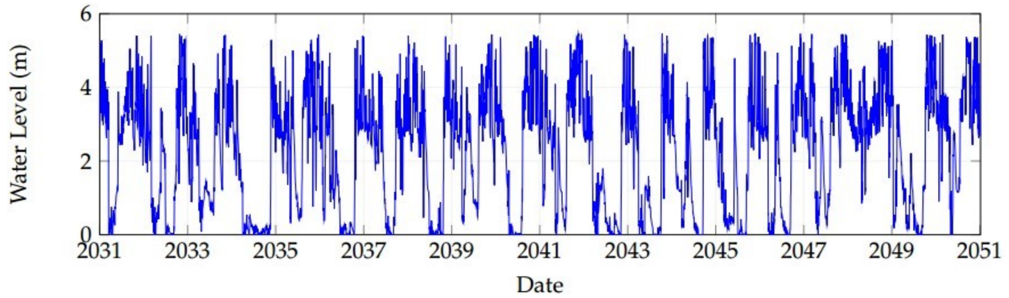
(a)



(b)



(c)



(d)

Figure 5.9: Projected future water levels: (a) for near-future at SSP2-4.5; (b) for near-future at SSP5-8.5; (c) for mid-future at SSP2-4.5; (b) for mid-future at SSP5-8.5

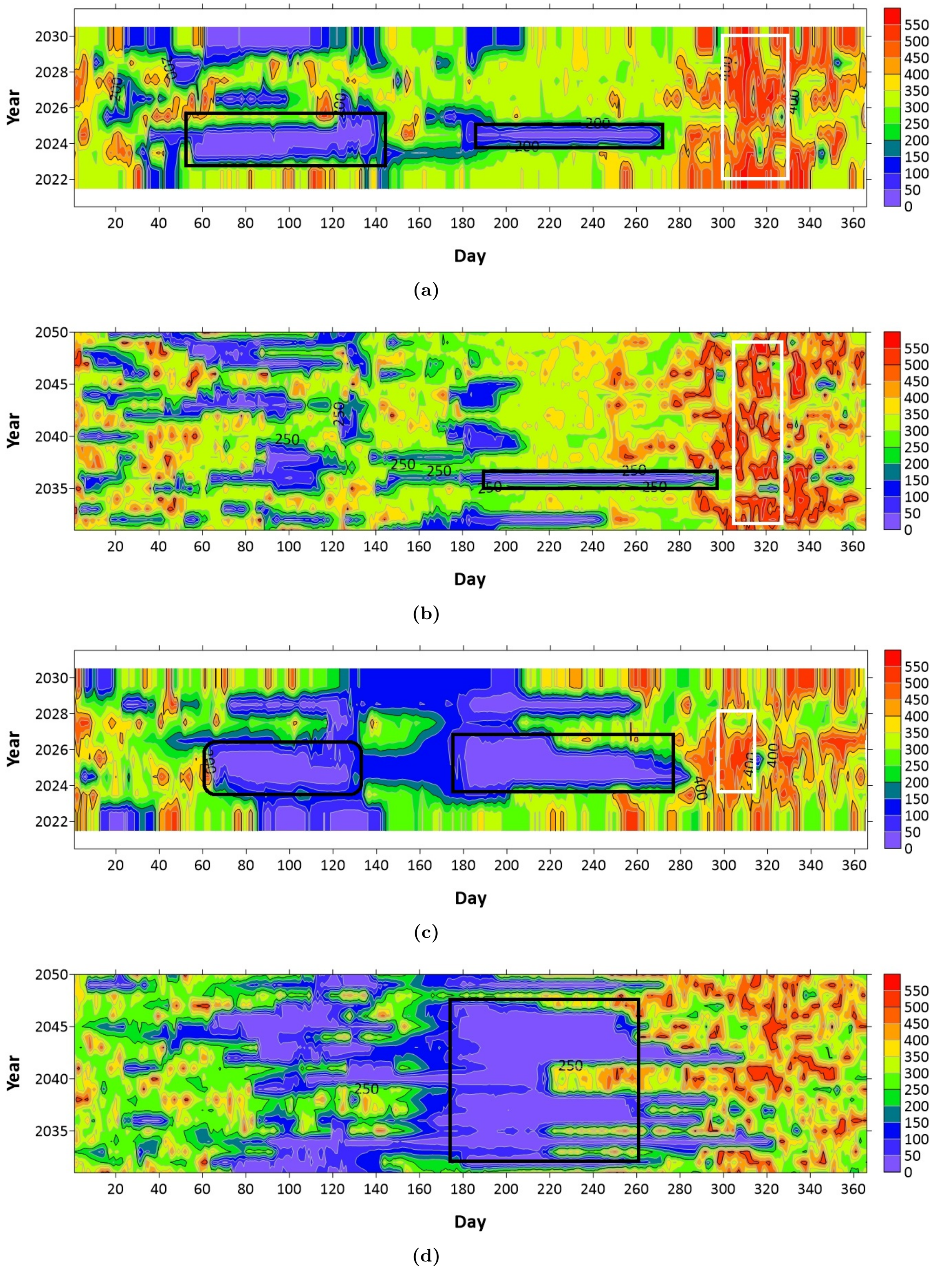


Figure 5.10: Water level predictions in a graphical way: (a) SSP2-4.5 for the year 2022 to 2030; (b) SSP5-8.5 for the year 2022 to 2030; (c) SSP2-4.5 for the year 2031 to 2050;(d) SSP5-8.5 for the year 2031 to 2050

Nevertheless, as discussed, more research is needed for a solid conclusion on future water levels.

5.6 Conclusions

An R-R prediction model was developed using the Cascaded-ANFIS algorithm for the Mahaweli River, the longest river in Sri Lanka. The R-R model was developed for the sub-catchment levels as well. The dataset used in the case study was well evaluated using four different methods of homogeneity tests Standard normal homogeneity test (SNHT), Buishand range (BR) test, Pettitt test, and von Neumann ratio (VNR) test. The algorithm was tested against three other regression algorithms used in most past studies: GRU, LSTM, and RNN. The results were comparatively studied using correlation coefficient, bias, RMSE, NSE, and KGE. The highest correlation coefficient was recorded by the Cascaded-ANFIS when utilizing the selected rainfall gauges to train the models having 0.933 where GRU, LSTM, and RNN showed the R values of 0.9133, 0.9120, and 0.8915, respectively.

Moreover, the bias value of the proposed algorithm is significantly low (1.52) compared with the other algorithms. The Cascaded-ANFIS model scored 0.66, 0.87, and 0.90 for RMSE, NSE, and KGE, respectively. These results outperformed the other algorithms used in this study.

According to the overall results, it can be concluded herein that the Cascaded-ANFIS algorithm-based prediction model has outperformed the other three algorithms. The second-best algorithm that performed well in prediction was the GRU algorithm. However, the Cascaded-ANFIS algorithm has advantages compared to the black-box regression models, such as lightweight, lower computational cost, easy real-time implementation, and efficiency. Therefore, the Cascaded-ANFIS algorithm can predict the water levels of various catchments under the requirement of measured rainfalls and water levels. More importantly, the model can be developed under mixed rainfall input along the timeline due to the upstream waterś travel time to the riverś downstream.

Overall the prediction model based on the Cascaded-ANFIS algorithm predicts accurate results using the ground-measured rainfalls. The water levels were projected under two SSP scenarios for the Manampitya station. However, promising results were only found under the near future and mid-future SSP rainfalls. None of the years was projected to have unacceptable floods (by looking at the records). Therefore, this research does not provide any conclusions about the future projected water levels.

References

[1] M. Amadori, V. Zamparelli, G. De Carolis, G. Fornaro, M. Toffolon, M. Bresciani, C. Giardino, and F. De Santi. Monitoring lakes surface water velocity with sar: A feasibility study on lake garda, italy. *Remote Sensing*, 13:2293, 2021.

[2] D. Floods Parker. (routledge. 2014.

[3] F. Chang, J. Liang, and Y. Chen. Flood forecasting using radial basis function neural networks. *IEEE Transactions On Systems, Man, And Cybernetics, Part C (Applications And Reviews)*, 31:530–535, 2001.

[4] R. Menezes and Others An. empirical analysis of mlp neural networks applied to streamflow forecasting. *IEEE Latin America Transactions*, 9:295–301, 2011.

[5] M. Romlay, M. Rashid, S. Toha, and A. Ibrahim. Rainfall-runoff model based on ann with lm, br and pso as learning algorithms. *International Journal Of Recent Technology And Engineering (IJRTE)*, 8:971–979, 2019.

[6] M. Rajurkar, U. Kothyari, and U. Chaube. Artificial neural networks for daily rainfall—runoff modelling. *Hydrological Sciences Journal*, 47:865–877, 2002.

[7] R. Vilanova, S. Zanetti, and R. Cecilio. Assessing combinations of artificial neural networks input/output parameters to better simulate daily streamflow: Case of brazilian atlantic rainforest watersheds. *Computers And Electronics In Agriculture*, 167, 2019.

[8] W. Wang, D. Xu, K. Chau, and S. Chen. Improved annual rainfall-runoff forecasting using pso–svm model based on eemd. *Journal Of Hydroinformatics*, 15:1377–1390, 2013.

[9] Z. Yaseen, M. Fu, C. Wang, W. Mohtar, R. Deo, and A. El-Shafie. Application of the hybrid artificial neural network coupled with rolling mechanism and grey model algorithms for streamflow forecasting over multiple time horizons. *Water Resources Management*, 32:1883–1899, 2018.

[10] O. Abiodun, A. Jantan, A. Omolara, K. Dada, N. Mohamed, and H. Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4, 2018.

[11] G. Corani and G. Guariso. Coupling fuzzy modeling and neural networks for river flood prediction. *IEEE Transactions On Systems, Man, And Cybernetics, Part C (Applications And Reviews)*, 35:382–390, 2005.

[12] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[13] B. Roy and M. Singh. An empirical-based rainfall-runoff modelling using optimization technique. *International Journal Of River Basin Management*, 18:49–67, 2020.

[14] V. Vidyarthi and S. Chourasiya. Particle swarm optimization for training artificial neural network-based rainfall–runoff model, case study: Jardine river basin. *Micro-Electronics And Telecommunication Engineering*, pages 641–647, 2020.

[15] M. Asadnia, L. Chua, X. Qin, and A. Talei. Improved particle swarm optimization–based artificial neural network for rainfall-runoff modeling. *Journal Of Hydrologic Engineering*, 19:1320–1329, 2014.

[16] R. Bensingh, R. Machavaram, S. Boopathy, and C. Jebaraj. Injection molding process optimization of a bi-aspheric lens using hybrid artificial neural networks (anns) and particle swarm optimization (pso). *Measurement*, 134:359–374, 2019.

[17] Q. Fang, H. Nguyen, X. Bui, T. Nguyen-Thoi, and J. Zhou. Modeling of rock fragmentation by firefly optimization algorithm and boosted generalized additive model. *Neural Computing And Applications*, 33:3503–3519, 2021.

[18] B. Jamali, M. Rasekh, F. Jamadi, R. Gandomkar, and F. Makiabadi. Using pso-ga algorithm for training artificial neural network to forecast solar space heating system parameters. *Applied Thermal Engineering*, 147:647–660, 2019.

[19] X. Bui, H. Nguyen, Q. Tran, D. Nguyen, and H. Bui. Predicting blast-induced ground vibration in quarries using adaptive fuzzy inference neural network and moth–flame optimization. *Natural Resources Research*, 30:4719–4734, 2021.

[20] H. Gholami, A. Mohamadifar, A. Sorooshian, and J. Jansen. Machine-learning algorithms for predicting land susceptibility to dust emissions: The case of the jazmurian basin, iran. *Atmospheric Pollution Research*, 11:1303–1315, 2020.

[21] J. Fan, L. Wu, F. Zhang, H. Cai, W. Zeng, X. Wang, and H. Zou. Empirical and machine learning models for predicting daily global solar radiation from sunshine duration: A review and case study in china. *Renewable And Sustainable Energy Reviews*, 100:186–212, 2019.

[22] I. Istadi and N. Amin. Modelling and optimization of catalytic–dielectric barrier discharge plasma reactor for methane and carbon dioxide conversion using hybrid artificial neural network—genetic algorithm technique. *Chemical Engineering Science*, 62:6568–6581, 2007.

[23] S. Nandi, Y. Badhe, J. Lonari, U. Sridevi, B. Rao, S. Tambe, and B. Kulkarni. Hybrid process modeling and optimization strategies integrating neural networks/-support vector regression and genetic algorithms: study of benzene isopropylation on hbeta catalyst. *Chemical Engineering Journal*, 97:115–129, 2004.

[24] A. Ahmad, I. Azid, A. Yusof, and K. Seetharamu. Emission control in palm oil mills using artificial neural network and genetic algorithm. *Computers & Chemical Engineering*, 28:2709–2715, 2004.

[25] A. Adib and A. Mahmoodi. Prediction of suspended sediment load using ann ga conjunction model with Markov chain approach at flood conditions. *KSCE Journal Of Civil Engineering*, 21:447–457, 2017.

[26] A. Mosavi, P. Ozturk, and K. Chau. Flood prediction using machine learning models: Literature review. *Water*, 10:1536, 2018.

[27] M. Shao, G. Zhao, S. Kao, L. Cuo, C. Rankin, and H. Gao. Quantifying the effects of urbanization on floods in a changing environment to promote water security—a case study of two adjacent basins in texas. *Journal Of Hydrology*, 589, 2020.

[28] K. Luo and X. Zhang. Increasing urban flood risk in china over recent 40 years induced by lucc. *Landscape And Urban Planning*, 219, 2022.

[29] G. Hollis. The effect of urbanization on floods of different recurrence interval. *Water Resources Research*, 11:431–435, 1975.

[30] Y. Bayazıt, C. Koç, and R. Bakış. Urbanization impacts on flash urban floods in bodrum province, turkey. *Hydrological Sciences Journal*, 66:118–133, 2021.

[31] B. Feng, Y. Zhang, and R. Bourke. Urbanization impacts on flood risks based on urban growth data and coupled flood models. *Natural Hazards*, 106:613–627, 2021.

[32] M. Noor, T. Ismail, E. Chung, S. Shahid, and J. Sung. Uncertainty in rainfall intensity duration frequency curves of peninsular malaysia under changing climate scenarios. *Water*, 10:1750, 2018.

[33] R. Bhatla, S. Verma, R. Pandey, and A. Tripathi. Evolution of extreme rainfall events over indo-gangetic plain in changing climate during 1901–2010. *Journal Of Earth System Science*, 128:1–14, 2019.

[34] J. Khan and M. Zaheer. Hydrological response to changing climate: a case study of large watershed, xianning, hubei, china. *Geology, Ecology, And Landscapes*, 6: 141–147, 2022.

[35] A. Prein, R. Rasmussen, K. Ikeda, C. Liu, M. Clark, and G. Holland. The future intensification of hourly precipitation extremes. *Nature Climate Change*, 7:48–52, 2017.

[36] A. Islam, A. Islam, G. Islam, S. Bala, M. Salehin, A. Choudhury, N. Dey, and A. Hossain. Adaptation strategies to increase water productivity of wheat under changing climate. *Agricultural Water Management*, 264, 2022.

[37] S. Wongchuig, J. Espinoza, T. Condom, H. Segura, J. Ronchail, P. Arias, C. Junquas, A. Rabatel, and T. A Lebel. *regional view of the linkages between hydro-climatic changes and deforestation in the Southern Amazon.* International Journal Of Climatology, 2022.

[38] K. Abass, G. Dumedah, F. Frempong, A. Muntaka, D. Appiah, E. Garsonu, and R. Gyasi. Rising incidence and risks of floods in urban ghana: Is climate change to blame? *Cities*, 121, 2022.

[39] L. Karamoutsou and A. Psilovikos. Deep learning in water resources management: The case study of kastoria lake in greece. *Water*, 13:3364, 2021.

[40] A. Sentas, A. Psilovikos, T. Psilovikos, and N. Matzafleri. Comparison of the performance of stochastic models in forecasting daily dissolved oxygen data in dam-lake thesaurus. *Desalination And Water Treatment*, 57:11660–11674, 2016.

[41] Avila Velasquez and Pulido-Velazquez D. *M. & Hector, M.* Improvement of water management for irrigation in Mediterranean basins combining remote sensing, weather forecasting, and artificial intelligence, 2022.

[42] X. Xie and D. Zhang. Data assimilation for distributed hydrological catchment modeling via ensemble kalman filter. *Advances In Water Resources*, 33:678–690, 2010.

[43] O. Kisi, J. Shiri, and M. Tombul. Modeling rainfall-runoff process using soft computing techniques. *Computers & Geosciences*, 51:108–117, 2013.

[44] R. Adnan, A. Petroselli, S. Heddam, C. Santos, and O. Kisi. Short term rainfall-runoff modelling using several machine learning methods and a conceptual event-based model. *Stochastic Environmental Research And Risk Assessment*, 35:597–616, 2021.

[45] F. Fahimi, Z. Yaseen, and A. El-shafie. Application of soft computing based hybrid models in hydrological variables modeling: a comprehensive review. *Theoretical And Applied Climatology*, 128:875–903, 2017.

[46] M. Rezaie-Balf, Z. Zahmatkesh, and S. Kim. Soft computing techniques for rainfall-runoff simulation: local non–parametric paradigm vs. model classification methods. *Water Resources Management*, 31:3843–3865, 2017.

[47] S. Chen, M. Ren, and W. Sun. Combining two-stage decomposition based machine learning methods for annual runoff forecasting. *Journal Of Hydrology*, 603, 2021.

[48] H. Herath, J. Chadalawada, and V. Babovic. Hydrologically informed machine learning for rainfall–runoff modelling: towards distributed modelling. *Hydrology And Earth System Sciences*, 25:4373–4401, 2021.

[49] B. A Mohammadi. review on the applications of machine learning for runoff modeling. *Sustainable Water Resources Management*, 7:1–11, 2021.

[50] V. Panwar and S. Sen. Disaster damage records of em-dat and desinventar: A systematic comparison. *Economics Of Disasters And Climate Change*, 4:295–317, 2020.

[51] J. Huizinga, H. De Moel, W. Szewczyk, and Others Global flood depth-damage functions. Methodology and the database with guidelines. (joint research centre (seville site). 2017.

[52] P. Arias, N. Bellouin, E. Coppola, R. Jones, G. Krinner, J. Marotzke, V. Naik, M. Palmer, G. Plattner, J. Rogelj, and Others Climate Change. The physical science basis. *Contribution of Working Group*, 14, 2021.

[53] K. Riahi, D. Vuuren, E. Kriegler, J. Edmonds, B. O'Neill, S. Fujimori, N. Bauer, K. Calvin, R. Dellink, O. Fricko, W. Lutz, A. Popp, J. Cuaresma, S. Kc, M. Leimbach, L. Jiang, T. Kram, S. Rao, J. Emmerling, K. Ebi, T. Hasegawa, P. Havlik, F. Humpen"oder, L. Silva, S. Smith, E. Stehfest, V. Bosetti, J. Eom, D. Gernaat, T. Masui, J. Rogelj, J. Strefler, L. Drouet, V. Krey, G. Luderer, M. Harmsen, K. Takahashi, L. Baumstark, J. Doelman, M. Kainuma, Z. Klimont, G. Marangoni, H. Lotze-Campen, M. Obersteiner, A. Tabeau, and M. Tavoni. The shared socioeconomic pathways and their energy, land use, and greenhouse gas emissions implications: An overview. *Global Environmental Change*, 42:153–168.

[54] J. Rogelj, A. Popp, K. Calvin, G. Luderer, J. Emmerling, D. Gernaat, S. Fujimori, J. Strefler, T. Hasegawa, G. Marangoni, V. Krey, E. Kriegler, K. Riahi, D. Vuuren, J. Doelman, L. Drouet, J. Edmonds, O. Fricko, M. Harmsen, P. Havlik, F. Humpen"oder, E. Stehfest, and M. Tavoni. Scenarios towards limiting global mean temperature increase below 1.5c. *Nature Climate Change*, 8:325–332.

[55] R. Samu and B. Akıntuğ. Pre-disaster planning and preparedness: drought and flood forecasting and analysis in zimbabwe. *Water Sa*, 46:448–457, 2020.

[56] A. Popp, K. Calvin, S. Fujimori, P. Havlik, F. Humpen"oder, E. Stehfest, B. Bodirsky, J. Dietrich, J. Doelmann, M. Gusti, and Others Land-use. futures in the shared socio-economic pathways. *Global Environmental Change*, 42:331–345, 2017.

[57] H. Schandl, Y. Lu, N. Che, D. Newth, J. West, S. Frank, M. Obersteiner, A. Rendall, and S. Hatfield-Dodds. Shared socio-economic pathways and their implications for global materials use. *Resources, Conservation And Recycling*, 160, 2020.

[58] M. Gidden, K. Riahi, S. Smith, S. Fujimori, G. Luderer, E. Kriegler, D. Vuuren, M. Berg, L. Feng, D. Klein, K. Calvin, J. Doelman, S. Frank, O. Fricko, M. Harmsen, T. Hasegawa, P. Havlik, J. Hilaire, R. Hoesly, J. Horing, A. Popp, E. Stehfest, and K. Takahashi. Global emissions pathways under different socioeconomic scenarios for use in cmip6: a dataset of harmonized emissions trajectories through the end of the century. *Geoscientific Model Development*, 12:1443–1475, 2019. URL `https://www.geosci-model-dev.net/12/1443/2019/`.

[59] M. Meinshausen, Z. Nicholls, J. Lewis, M. Gidden, E. Vogel, M. Freund, U. Beyerle, C. Gessner, A. Nauels, N. Bauer, and Others The. shared socio-economic pathway (ssp) greenhouse gas concentrations and their extensions to 2500. *Geoscientific Model Development*, 13:3571–3605, 2020.

[60] S. Dangol, R. Talchabhadel, and V. Pandey. Performance evaluation and bias correction of gridded precipitation products over arun river basin in nepal for hydrological applications. *Theoretical And Applied Climatology*, 148:1353–1372, 2022.

[61] T. Lafon, S. Dadson, G. Buys, and C. Prudhomme. Bias correction of daily precipitation simulated by a regional climate model: a comparison of methods. *International Journal Of Climatology*, 33:1367–1381, 2013.

[62] I. Chathuranika, M. Gunathilake, H. Azamathulla, and U. Rathnayake. Evaluation of future streamflow in the upper part of the nilwala river basin (sri lanka) under climate change. *Hydrology*, 9:48, 2022.

[63] M. De Silva and A. Kawasaki. *Modeling the association between socioeconomic features and risk of flood damage: A local-scale case study in Sri Lanka.* Risk Analysis, 2022.

[64] T. Sirisena, S. Maskey, J. Bamunawala, E. Coppola, and R. Ranasinghe. Projected streamflow and sediment supply under changing climate to the coast of the kalu river basin in tropical sri lanka over the 21st century. *Water*, 13:3031, 2021.

[65] S. Shelton and Z. Lin. Streamflow variability in mahaweli river basin of sri lanka during 1990–2014 and its possible mechanisms. *Water*, 11:2485, 2019.

[66] S. Withanachchi, S. K"opke, C. Withanachchi, R. Pathiranage, and A. Ploeger. Water resource management in dry zonal paddy cultivation in mahaweli river basin, sri lanka: An analysis of spatial and temporal climate change impacts and traditional knowledge. *Climate*, 2:329–354, 2014.

[67] S. Chandrasekara, V. Prasanna, and H. Kwon. Monitoring water resources over the kotmale reservoir in sri lanka using enso phases. *Advances In Meteorology*, 2017, 2017.

[68] R. Taylor. Interpretation of the correlation coefficient: a basic review. *Journal Of Diagnostic Medical Sonography*, 6:35–39, 1990.

[69] J. Nash and J. Sutcliffe. River flow forecasting through conceptual models part i—a discussion of principles. *Journal Of Hydrology*, 10:282–290, 1970.

[70] H. Gupta, H. Kling, K. Yilmaz, and G. Martinez. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling. *Journal Of Hydrology*, 377:80–91, 2009.

Chapter 6

Application of Generic rainfall-runoff model implementation

This work is currently (update 2022/12/21) under review in Natural Resources Research (Springer) journal.

6.1 Introduction

River flows and flow-related analyses such as floods are complex and challenging due to natural geographical circumstances, the combined effect of rainfall occurrences, and catchment basin characteristics[1]. Many sophisticated hydrologic and hydraulic models are available to analyze the flow scenarios, and their impact on the nearby flood plains [2, 3]. However, these numerical models require several inputs, including precipitation, atmospheric conditions, soil, other geological information, friction parameters, geometrical features of rivers, etc. [4, 5]. Therefore, these input parameters highly influence the accuracy of the outcome, which is the flow situation downstream. In addition, some of these required parameters are Spatio-temporal dependent [6]. Thus, the analysis becomes complicated. However, recent advances in computational competencies have analyzed these complex scenarios attractive [7, 8].
Nevertheless, there is still a scope to improve the accuracy of river flow prediction in the context of soft computing techniques and machine learning approaches. Robust

nonlinear regression approaches are required to represent the river flows and their related research work; thus, artificial intelligence (AI) and soft computing approaches are competent in handling these complexities. Genetic Programming (GP), Support Vector Machine (SVM), and Multigene Genetic Programming (MGP) are some of the soft computing approaches used in the literature to predict river flows [9, 10]. In addition, Artificial Neural Network (ANN) based and Fuzzy Logic (FL) based river flow and flood forecasting models can be identified in the literature in the context of AI to the flow predictions. Historical rainfall and runoff measurements are routinely used to mimic the process [11]. However, AI approaches can estimate nonlinear functions to establish a precise connection between the state parameters of the system without making explicit assumptions about the theoretical aspects of flow situations. As a result, a detailed understanding of hydrological processes is unnecessary for AI-based R-R modelling. Tokar and Johnson [12] had presented an ANN approach to predict runoff based on rainfall and found interesting results when the model was applied to the Little Patuxent River watershed in Maryland, USA. Their approach has received accurate results while maintaining the model's calibration flexibility. In addition, the processing time was quick for calibration. Many researchers applied and tested similar or modified algorithms and found accurate results for runoff modelling by ANN. Senthil Kumar et al. [13], Srunivasulu and Jain [14], Machado et al. [15], and Wu and Chau [16] are some of the examples of such studies which can be found in the literature.

In addition, some researchers have developed hybrid models to improve the effectiveness of AI [17]. Wavelet Transform (WT), Principal Component Analysis (PCA), singular spectrum analysis (SSA), and moving average (MA) are some examples to showcase the hybrid approaches [18–21]. Hu et al. [18] demonstrated that PCA using delayed data could improve the accuracy of the ANN approach for the flood forecasting process when applied to the Darong River basin, China. According to Wu and Chau [16], SSA is an excellent strategy for removing the lagging prediction influence of ANN algorithms. In addition, some researchers have used hybrid models to predict the runoff combing Dynamic Artificial Neural Networks (DANN) and Wavelet-Artificial Neural Networks (WANN) [Sharghi et al. [22], and Nourani et al. [23]. Nourani et al. [19] used discrete WT to study flood forecasting data from the Lighvan Chai catchment in Iran. The results showed that WANN performs better than the traditional ANN model for flood forecasting. In addition, Nourani et al. [24] and Pramanik et al. [25] demonstrated the efficacy and capacity to overcome the shortcomings of standalone models using the

WANN approach. However, De Vos and Rientjes [20] found that lagged prediction had negatively influenced ANN model performance. This was due to the autoregressive connection between forecasting model inputs. As a result, a hybrid MA-ANN model was developed using an MA data pre-processing method, outperforming the standalone ANN model. Wu et al. [21] proved the efficiency of pre-processing of the MA data approach for daily streamflow estimate in comparative research. Chang et al. [26] have proposed a forecasting model based on a neural network with radial basis function (RBF). They have used the fuzzy clustering method to determine the nonlinear RBF. Moreover, a neural network and fuzzy logic-based flood prediction system are proposed by Corani et al. [27]. They employed fuzzy membership functions to classify the saturation state as the initial stage. Then use neural networks for the prediction of floods.

Some other researchers have developed hybrid techniques using ANN and GP. Shoaib et al. [28] suggested a model using a hybrid Wavelet Gene Expression Programming (WGEP) with several datasets. The WGEP strategy outperformed a standalone GEP technique, according to the findings. In addition, Nourani et al. [29] investigated the efficacy of an Emotional Artificial Neural Network (EANN) for flood forecasting in two distinct catchments. EANN outperforms better when compared with traditional ANN algorithms in the stability of the EANN strategy in determining dry and wet environments using hormonal factors. Sharghi et al. [22] evaluated the performance of traditional ANN and EANN with WANN techniques in two rivers from different locations. The results demonstrate the usefulness of the EANN and WANN approaches for flood forecasting. Nourani et al. [30] have further used the WT to partition the past flood forecasting time series into several elements in two Iranian watersheds, resulting in hybrid GP-ANN models. On the other hand, FL techniques significantly used river flow and flood prediction [31]. Hundecha et al. [32] have investigated rainfall-runoff behaviour using routine-based fuzzy rules. They tested the model in the Neckar River catchment in southwest Germany and found accurate results. Özelkan and Duckstein [33] has developed a conceptual R-R model using FL to estimate the various parameters required for R-R modelling. They have applied this developed model to Walnut Gulch experimental watersheds in Arizona and obtained more stable parameter estimations. Şen and Altunkaynak [34] applied a comparative FL approach to identify the runoff coefficients and then estimate the runoff of two basins on the European and Asian sides of Istanbul.

In addition, Tayfur and Brocca [35], and Casper et al. [36] have developed R-R models based on soil moisture and found exciting results. Chang et al. [37] developed an ANFIS flood forecasting of a catchment using two input selection techniques: Mutual Information and Cross-Correlation Analyses (MICCA) and Cross-Correlation Analysis (CCA). Using rainfall as input parameters, they have found comparable results to one of the widely used hydrologic models, HEC-HMS. Furthermore, Chang et al. [35] have further developed the resilience of the FL alternatives using the Self-adaptive Fuzzy Inference Network (SaFIN) and Adaptive Network-Based Fuzzy Inference System (ANFIS) in the context of flood forecasting when the data contains instabilities.

However, due to its purely nonlinear behaviour, improved computational techniques are essential to enhance the accuracy of the R-R modelling. The enhanced accuracy would help plan and manage the water systems sustainably while catering to water scarcity and minimizing flood damages under the minimum input data. Therefore we introduce a novel, highly accurate, and efficient R-R modelling approach using the Cascaded-ANFIS algorithm. The research impact of this study is promising. The novel approach is less computationally complex as it needs only rainfall as the input variable. Thus, the real-time application is possible with low computational power while ensuring less resource utilization. The developed R-R modelling approach was compared against five state-of-the-art algorithms used in the research world and stated its high accuracy. In addition, the approach is generic and can be applied to any river system and proved its performance using several river systems in Japan, Vietnam, and Sri Lanka.

6.2 Methodology

6.2.1 Algorithm Development

As stated earlier, this research aims to develop a relationship between the runoff and rainfall of a catchment. Mathematically, this relationship can be formulated as Equation 6.1. Cascaded-ANFIS algorithms were used to develop the nonlinear function stated in the Equation.

$$Runoff = Function(Rainfall_i) \tag{6.1}$$

The structure of the Cascaded-ANFIS can be modified according to the problem, which is one of the significant benefits. The developed structure of the algorithm is organized as shown in Figure 6.1. This study focuses more on a few input features, including daily rainfall data of the catchment area. As shown in this figure, the number of input parameters was selected as six, and they are named Rainfall ST_n where n is the corresponding number of the rainfall measuring station.

The overall system implementation can be explained as follows. First, the dataset was initialized. These data were collected from well-known dataset repositories. The dataset is generally a combination of inputs and outputs. In this study, the rainfall measurements were considered the inputs, and the output was the river's water level at the desired place. Each data set was divided into two clusters; the training set and the validation set. The proportions were kept at 70% and 30%, respectively. As introduced in the above subsection, the Cascaded-ANFIS has two main modules: pair selection and training. Therefore, the input data was fed into the pair selection section. Then each input was paired with the most suitable other input and converged to the next stage, training the model. As in Figure 6.1, there are six pairs in the system, and they were being handled individually by a two-input one-output ANFIS module. This operation provides two outputs: RMSE and predicted value. According to the system implementation, the increase in the Cascaded-ANFIS can be dealt with in two methods: pre-determined RMSE or the maximum iterative levels. This study has considered the pre-determined levels since the problem statement is a regression model. At each level, the outputs were re-routed as input to the system. Figure 6.1 contains six outputs: Output n, where n is the corresponding ANFIS module. When converging to the second level from the first level, these outputs were used as the inputs and repeatedly fed into the pair selection section. Then the second level continues. However, once the decided maximum iteration level is reached, the outputs are averaged, and the final prediction is obtained (Equation 6.2).

$$F = \frac{\sum_{i=1}^{n} O_n}{n} \tag{6.2}$$

The developed Cascaded-ANFIS algorithm was applied to several river basins in different countries and tested for the algorithm's robustness. The R-R model was tested for three combinations of rainfalls and runoff. These scenarios are given in the Equations below. Scenario 1 (Equation 6.3)showcases the rainfall and runoff on the same day, whereas scenario 2 (Equation 6.4) analyses the runoff based on the previous day's rainfall. Scenario 3 (Equation 6.5)analyses the runoff with rainfall which happened two days ago.

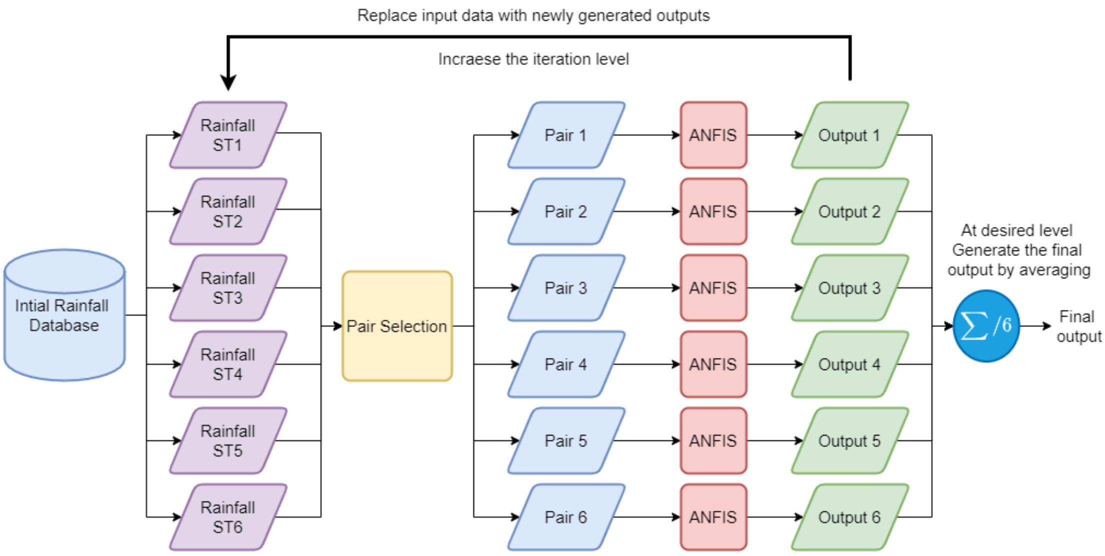


Figure 6.1: Rainfall-runoff model implementation flowchart using Cascaded-ANFIS algorithm

$$Runoff_i = Function(Rainfall_{i,t}) \tag{6.3}$$

$$Runoff_i = Function(Rainfall_{i,t-1}) \tag{6.4}$$

$$Runoff_i = Function(Rainfall_{i,t-2}) \tag{6.5}$$

6.2.2 Performance evaluation metrics

The algorithm's performance was then tested by several metrics, including root mean square error (RMSE), bias, Nash-Sutcliffe efficiency (NSE), Kling- Gupta Efficiency (KGE), and correlation coefficient (R). These performance evaluating metrics are given in Equations 7.6, 7.2, 7.3, and 7.1.

$$RMSE = \sqrt{\frac{1}{q}\sum_{t=1}^{q}(u(t) - \bar{u}(t))^2} \tag{6.6}$$

$$bias = \frac{\sum_{j=1}^{k} u(t) - \bar{u}(t)}{\sum_{j=1}^{k} u(t)} \tag{6.7}$$

$$NSE = 1 - \frac{\sum_{j=1}^{k}(u(t) - \bar{u}(t))^2}{\sum_{j=1}^{k}(u(t) - \bar{v}(t))^2} \tag{6.8}$$

$$R = \frac{\sum(v(t) - \bar{v}(t))(u(t) - \bar{u}(t))}{\sqrt{\sum(v(t) - \bar{v}(t))^2 \sum(u(t) - \bar{u}(t))^2}} \tag{6.9}$$

where $u(t)$ is the predicted parameter, $\bar{u}(t)$ is the mean of predicted parameter$v(t)$ is the measured parameter, k is the population size and $\bar{v}(t)$ is the mean of measured parameter. The correlation coefficient (R) represents the goodness of fit. It varies from -1 to 1; the best is when it becomes 1. Bias tells the differences between predicted to measured values. The ideal bias value is 0, and 1 becomes the worst. NSE calculates the perfectness of the match between real and prediction. The results of the NSE can vary between minus infinity being the worst and 1 being the ideal [38].KGE is a combined calculation of three main parameters: NSE, bias, and coefficient of variation. Recently it has been used more in hydrological model performance calculations [39].

6.2.3 Parameter Settings

This section introduces the parameters used in the algorithms used in this study for the comparative analysis. Six state-of-the-art algorithms were used along with the Cascaded-ANFIS for the examination. They can be pointed out as follows.

1. Lasso Regression
2. Linear Regression
3. Long Short-Term Memory (LSTM)
4. Grated Recurrent Unit (GRU)
5. Recurrent Neural Network (RNN)
6. Cascaded Adaptive Network-Based Fuzzy Inference System (Cascaded-ANFIS)

These ML algorithms were considered in this study due to a few specific reasons, such as algorithms being similar and easy implementation. Moreover, they are low-weight and can be processed in a general computer without GPU support.
Table 10.1 shows that the same parameter values were considered for LSTM, GRU, and RNN turning. The Cascaded-ANFIS used three Gaussian membership functions for each input in the system. The whole cascades were 30 to achieve satisfactory accuracy and error value.

Table 6.1: Parameter settings of the algorithms used in this study.

Algorithm	Parameters	
RNN/LSTM/GRU	Optimizer	adam
	Learning rate	0.0001
	Activation	relu
	Batch size	72
	Epoches	1000
Cascaded-ANFIS	Iterations	100
	Membership Functions	Gausian
	Number of Membership Functions	3
	Number of Cascades	30
	Step Size	0.1
	Decrease rate	0.9
	Increase rate	1.1

6.2.4 Evaluation Benchmarks

The evaluation of the proposed algorithm was conducted by employing state-of-the-art machine learning algorithms such as Recurrent Neural Network (RNN), Linear Regression (LIRE), Ridge Regression (RIRE), Lasso Regression (LARE), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU).therefore, the results of each river basin can be checked for the accuracy of the predicted variables.

6.3 Applications of Cascaded-ANFIS stream flow prediction

The developed R-R prediction algorithm was applied to five major rivers in Japan, Vietnam, and Sri Lanka. The following sections present the details of these five river basins.

6.3.1 Monobe River, Japan

The Monobe River is one of the main rivers in Shikoku Island, Japan. It starts at Mount Akagiouama, which is 1436 m in height and is located in the Tsurugi mountain range. The total length of the river can be approximated as 71 km. The river has a catchment area of 508.2 km^2 which combines three sections of 8.2 km^2 of a watercourse, 461.8 km^2 of a forest, and 38.2 km^2 of a flat land. Figure 6.2 shows the catchment area of Monobe River with its selected rain gauges and the water level measuring station. The average discharge of the river is around 24.6 m^3/s. The discharge is highly dependent on the

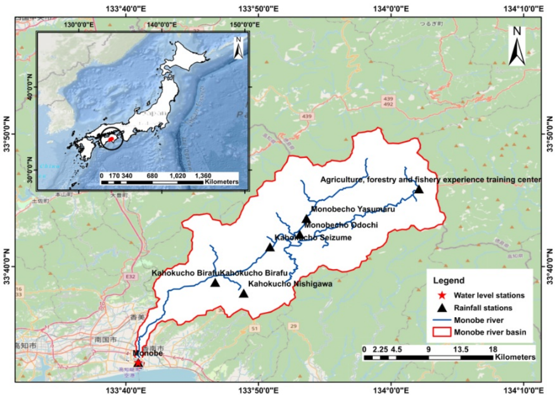


Figure 6.2: Monobe River rainfall catchment map

rainfall, as inspected from the dataset. Daily rainfall data for eight rain gauges from 2010 to 2019 were obtained from Japan Meteorological Agency (3058 data points). In addition, the temporal variation of stage measurements (to showcase the discharge) was obtained from the same agency. The rainfalls are the inputs and are given in millimetres, whereas the stages (water levels) are the outputs and are given in centimetres.

More descriptive information is given in Appendix A Table A.1. The basic statistical information is given in the table for each station as means and standard deviations. As expected, higher standard deviations can be seen due to the temporal scale's non-uniformity of rainfall and water levels. The river's water level has a wider range, from 7 cm as the minimum and 263 cm as the maximum. The distribution of the dataset is presented by using the 25%, 50%, and 75% for all rainfall stations and the water level. Herein, the sample value for each portion is given accordingly. Moreover, the mean value (Mean), standard deviation (STD), and max value (Max) are shown in the table for each variable of the dataset.

6.3.2 Niyodo River, Japan

Niyodo River is another main river in Shikoku Island, Japan. It starts at Mount Ishizuchi, which is 1982 m in height and is located on the border of Saijō and Kumakōgen, Ehime, Japan. Figure 6.3 showcases the catchment area with 14 rain gauges and a water level measuring station. The total length of the river is approximated as 124 km. The river is enriched with a basin area of 1560 km^2. Four main bridges exist along the river: Nagoyachinka Bridge, Niyodogawa Estuary Bridge, Kataokachinka Bridge, and Odo Dam Bridge. The average discharge of the Monobe River is around 100 m^3/s.

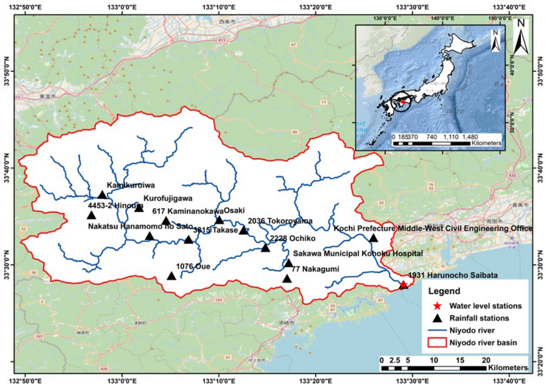


Figure 6.3: Niyodo River rainfall catchment map

Daily rainfall data (in mm) and river water level data (in cm) were obtained from Japan Meteorological Agency from 2010 to 2019.

Appendix A Table A.2 showcases the basic statistical information of data obtained. The maximum water level observed in this river is almost 10 m (999 cm). The distribution of the dataset is presented by using the 25%, 50%, and 75% for all rainfall stations and the water level. Herein, the sample value for each portion is given accordingly. Moreover. The mean value (Mean), standard deviation (STD), and max value (Max) are shown in the table for each variable of the data set.

6.3.3 Thu Bon River, Vietnam

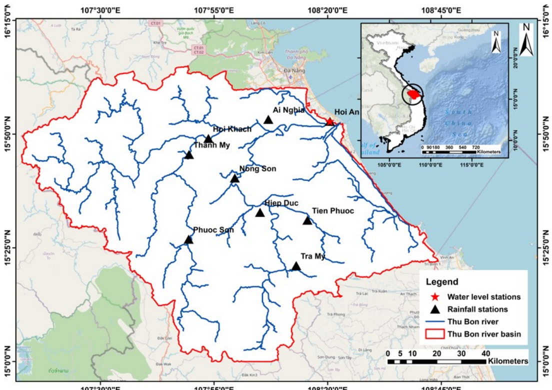


Figure 6.4: Thu Bon River rainfall catchment map

Vietnam is ranked eighth in the world for extreme weather events. Generally, the Thu Bon River floods annually from October to December due to the rainy season in Vietnam [40]. Therefore, this river was selected as one of the applications for the novel Cascaded-ANFIS algorithm. Thu Bon River is one of the largest rivers in Vietnam. The total length of this river is around 205 km. It starts at a mountain called Ngoc Linh, "The

roof of southern Vietnam". The endpoint of the river is the South China Sea. The Thu Bon diver is situated in Quang Nam province in southern Vietnam. Figure 6 presents the catchment of the Thu Bon River with the locations of the nine rain gauges and one water level measuring station. The daily rainfall data and water levels were obtained from the Vietnam Metrology Institute from 2003 to 2007. The total catchment area is roughly 10000 km^2, and the river's average discharge is about 9100 m^3/s.
The overall dataset descriptive analysis is shown in Appendix A Table A.3. The distribution of the dataset is presented by using the 25%, 50%, and 75% for all rainfall stations and the water level. Herein, the sample value for each portion is given accordingly. Moreover. The mean value (Mean), standard deviation (STD), and max value (Max) are shown in the table for each variable of the data set.

6.3.4 Kelani River, Sri Lanka

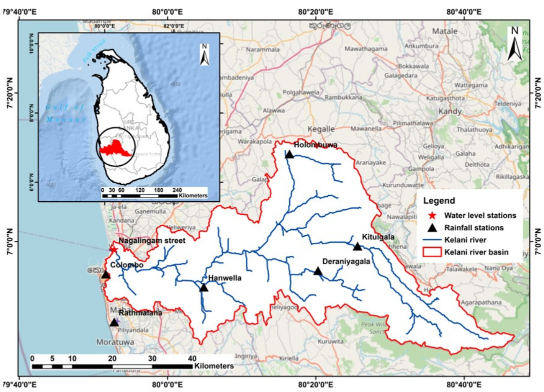


Figure 6.5: Kelani River rainfall catchment map

Sri Lanka has many rivers (103), and thus, flooding is more common. The wet zone of Sri Lanka is annually flooded during the southwestern monsoon (May to September). Therefore, the Kelani River, which is flowing via the commercial capital of the country (Colombo), was selected as a test study for this research. The Kelani River is the fourth-longest river in Sri Lanka. It is 145 km long and has a catchment area of 2340 km^2. It begins from Adams Peak in the central mountainous region, travels over steep slopes, and enters a gentle sloping intermediate and flat coastal plain before reaching the Indian Ocean. As a result of the river catchment's location in the wet zone, it receives an annual average rainfall of 3450 mm. The catchment is complex and contains 20 sub-catchments comprised of eleven landforms [41]. The Kelani River provides 80% of Greater Colombo's water supply. Therefore, the importance of analyzing river flow is highly stated. Figure

6.5 presents the Kelani River basin's catchment area and gauging stations. The daily rainfall data from 2003 to 2017 were purchased from the Meteorological Department of Sri Lanka, while the daily water levels for the same period were purchased from the Department of Irrigation, Sri Lanka.

In recent years, the frequency and severity of extreme floods have significantly increased in the Kelani River basin. Thus, fatal and property damages are reported [42]. The Department of Irrigation, Sri Lanka, noted the severity of flooding in the low-lying area of the Kelani River based on the gauge post-reading of the Nagalagam Street station in Colombo. If the river flows at Nagalagam Street could be precisely predicted, the downstream of Hanwella, including the heavily populated City of Colombo, might be adequately protected. Appendix A Table A.4 shows the descriptive analysis of the dataset used in this study for the Kelani River case study. The distribution of the dataset is presented by using the 25%, 50%, and 75% for all rainfall stations and the water level. Herein, the sample value for each portion is given accordingly. Moreover. The mean value (Mean), standard deviation (STD), and max value (Max) are shown in the table for each variable of the data set.

6.3.5 River 1 - Kalu River (Sri Lanka)

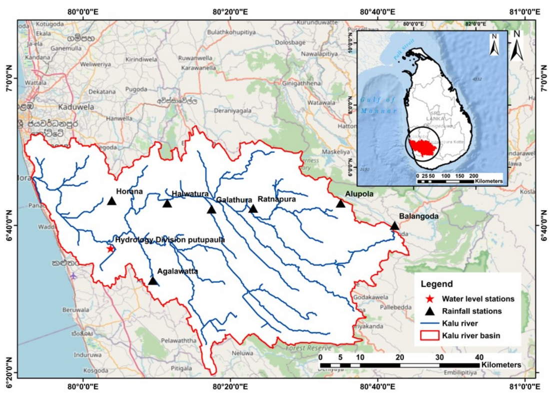
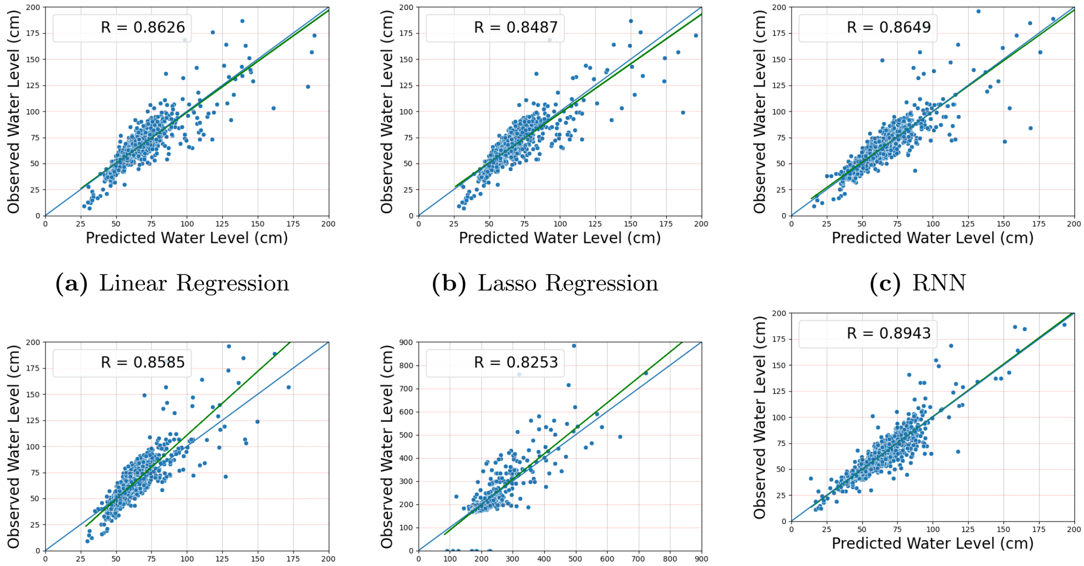


Figure 6.6: Kalu River rainfall catchment map

The Kalu River basin is the second largest river basin in Sri Lanka, encompassing an area of 2766 km^2, with most of its catchment lying in the region that gets the most annual precipitation in the country. The average annual rainfall in the basin is 4000 mm [43]. Although the Kalu River has the second-largest catchment area in the country, it releases the most volume of water to the sea, around four billion m^3/year [42] [43]. Figure 6.6 shows the catchment area with other gauging stations. The Kalu River starts



(a) Linear Regression **(b)** Lasso Regression **(c)** RNN

(d) LSTM **(e)** GRU **(f)** Cascaded-ANFIS(20)

Figure 6.7: Comparison of the Correlation Coefficient (R^2) - Monobe River in japan

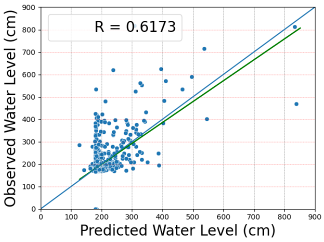
in the central highlands in the wet zone at an elevation of 2250 m above Mean Sea Level (MSL), flows through the western slopes and then the western plains before emptying into the sea Kalutara after a distance of around 129 km. Upper basin areas have high slopes, whereas lower basin areas have gentle gradients. Due to the hydrological and geological characteristics of the river basin, the Ratnapura area and its middle and lower flood plain frequently flooded during the southwest monsoon season. However, these areas are highly urbanized. Thus, accurate river flow prediction is highly important. Therefore, the novel algorithm was tested for this river basin. The daily rainfall data from 2000 to 2015 were purchased from the Meteorological Department of Sri Lanka, while the daily water levels for the same period were purchased from the Department of Irrigation, Sri Lanka. Appendix A Table A.5 shows the Kalu River dataset descriptive analysis.

6.4 Results and Discussion

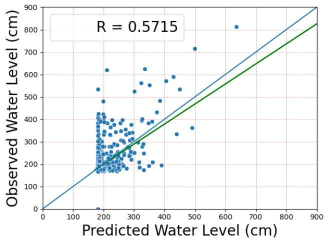
6.4.1 R-R model for the Monobe River

Figure 6.7 presents the correlation of predicted water level to ground-measured water level for the Monobe River under different algorithms.

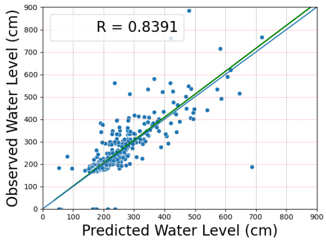
The proposed algorithm performed the best having 0.8943 as the R-value, while the RNN scored 0.8626 as the second best. The GRU model showed the lowest R-value



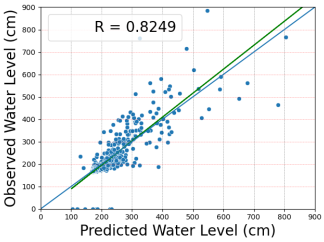
(a) Linear Regression



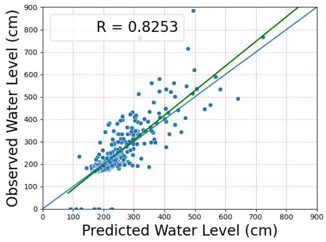
(b) Lasso Regression



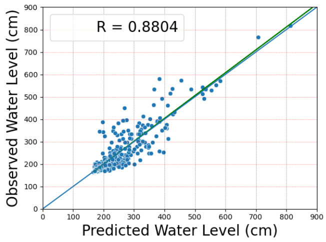
(c) RNN



(d) LSTM

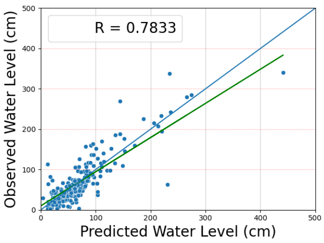


(e) GRU

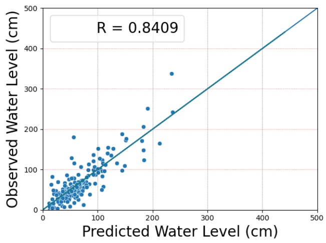


(f) Cascaded-ANFIS(20)

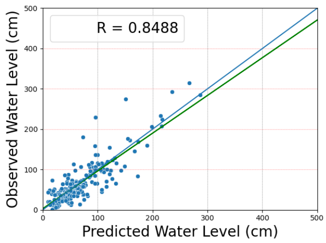
Figure 6.8: Comparison of the Correlation Coefficient (R^2) - Niyodo River in japan



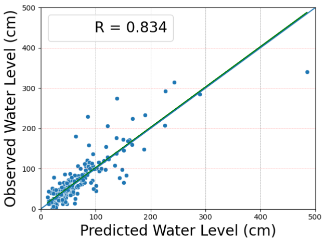
(a) Linear Regression



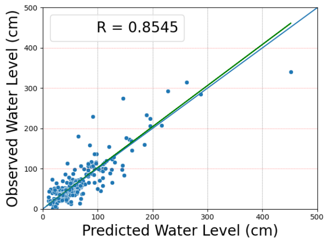
(b) Lasso Regression



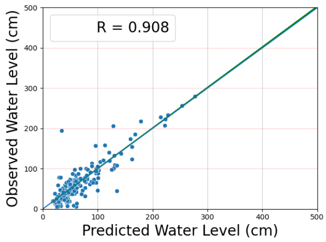
(c) RNN



(d) LSTM

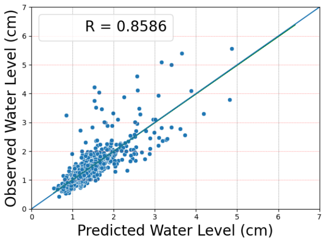


(e) GRU

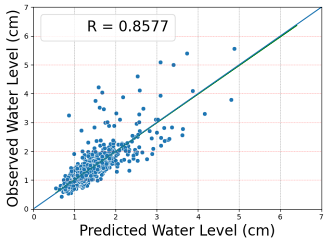


(f) Cascaded-ANFIS(20)

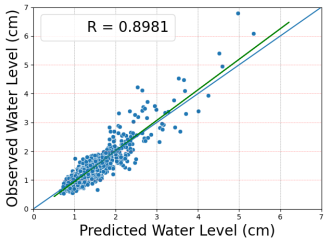
Figure 6.9: Comparison of the Correlation Coefficient (R^2) - THU BON Rive in Vietnam



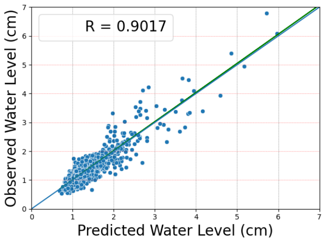
(a) Linear Regression



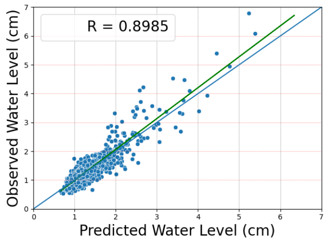
(b) Lasso Regression



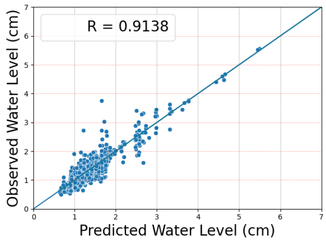
(c) RNN



(d) LSTM

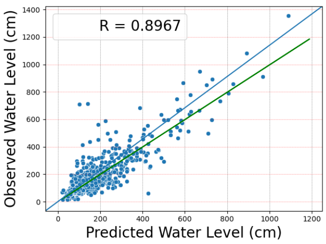


(e) GRU

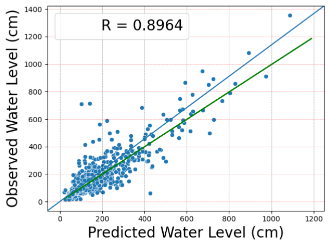


(f) Cascaded-ANFIS(30)

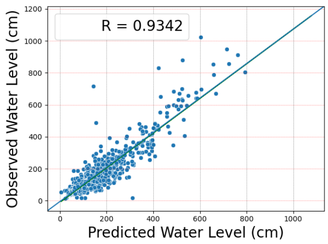
Figure 6.10: Comparison of the Correlation Coefficient (R^2) - Kelani River in Sri Lanka



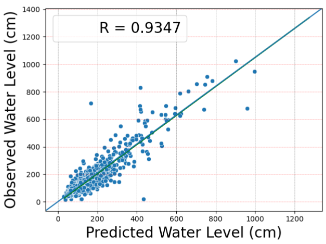
(a) Linear Regression



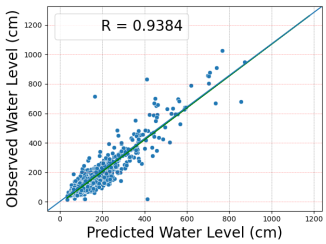
(b) Lasso Regression



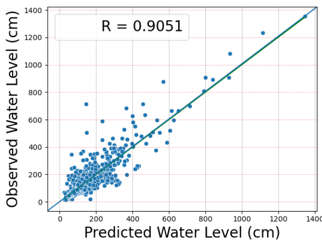
(c) RNN



(d) LSTM



(e) GRU



(f) Cascaded-ANFIS(30)

Figure 6.11: Comparison of the Correlation Coefficient (R^2) - Kalu River in Sri Lanka

having 0.8253. The total number of iterations used for the Cascaded-ANFIS was 30 for the Monobe River dataset. Therefore, it can be seen that the correlation for Cascased-ANFIS outperformed all other algorithms. Thus, the developed algorithm can produce accurate results compared to other state-of-the-art algorithms. Appendix A Figure A.1 further justifies this finding. As shown in the figure, the bias of the Cascaded-ANFIS is zero, while other algorithm shows a significant increase or decrease in bias. NSE and KGE values of the Cascaded-ANFIS outperform the other algorithms by having 0.80 and 0.84, respectively. Moreover, the proposed algorithm also shows the least RMSE of 9.46.

6.4.2 R-R model for Niyodo River

Figure 6.8 presents the correlation of predicted water level to ground-measured water level for the Niyodo River under different algorithms.

The proposed algorithm performed the best having 0.8804 as the R-value, while the RNN scored 0.8391 as the second best. The Lasso Regression model showed the lowest R-value having 0.5715. The total number of iterations used for the Cascaded-ANFIS was 30 for the Niyodo River dataset. Therefore, it can be seen that the correlation for Cascased-ANFIS outperformed all other algorithms. Therefore, the developed algorithm can produce accurate results compared to other state-of-the-art algorithms. Appendix A Figure A.2 further justifies this finding. As shown in the figure, the bias of the Cascaded-ANFIS is zero, while other algorithm shows a significant increase or decrease in bias. NSE and KGE values of the Cascaded-ANFIS outperform the other algorithms by having 0.77 and 0.82, respectively. Moreover, the proposed algorithm also shows the least RMSE of 35.51.

6.4.3 R-R model for Thu Bon River

Figure 6.9 presents the correlation of predicted water level to ground-measured water level for the Thu Bon River under different algorithms.

The proposed algorithm performed the best having 0.9080 as the R-value, while the GRU scored 0.8545 as the second best. The Linear Regression model showed the lowest R-value having 0.7833. The total number of iterations used for the Cascaded-ANFIS was 30 for the Thu Bon River dataset. Therefore, it can be seen that the correlation for Cascased-ANFIS outperformed all other algorithms. Therefore, the developed algorithm can produce accurate results compared to other state-of-the-art algorithms. Appendix A Figure A.3 further justifies this finding. As shown in the figure, the bias of the Cascaded-ANFIS is zero, while other algorithm shows a significant increase or decrease in bias. NSE and KGE values of the Cascaded-ANFIS outperform the other algorithms by having 0.82 and 0.87, respectively. Moreover, the proposed algorithm also shows the least RMSE of 22.17.

6.4.4 R-R model for Kelani River

Figure 6.10 presents the correlation of predicted water level to ground-measured water level for the Kelani River under different algorithms.
The proposed algorithm performed the best having 0.9138 as the R-value, while the LSTM scored 0.9017 as the second best. The Lasso Regression model showed the lowest R-value having 0.8577. The total number of iterations used for the Cascaded-ANFIS was 30 for the Kelani River dataset. Therefore, it can be seen that the correlation for Cascased-ANFIS outperformed all other algorithms. Therefore, the developed algorithm can produce accurate results compared to other state-of-the-art algorithms. Appendix A Figure S4 further justifies this finding. As shown in the figure, the bias of the Cascaded-ANFIS is zero, while other algorithm shows a significant increase or decrease in bias. NSE and KGE values of the Cascaded-ANFIS outperform the other algorithms by having 0.84 and 0.88, respectively. Moreover, the proposed algorithm also shows the least RMSE of 0.21.

6.4.5 R-R model for Kalu River

Figure 6.11 presents the correlation of predicted water level to ground-measured water level for the Kalu River under different algorithms.
The proposed algorithm scored 0.9051 as the R-value, while the GRU scored 0.9384 as the best R-score. The Lasso Regression model showed the lowest R-value having 0.8964. The total number of iterations used for the Cascaded-ANFIS was 30 for the Kalu River dataset. Although, in this case, the R-score of the proposed algorithm is lower than the LSTM, GRU, and RNN, it can be seen that the deference of correlation for Cascased-ANFIS between other algorithms is much smaller. Therefore, the developed algorithm can produce accurate results compared to other state-of-the-art algorithms. Appendix A Figure S2 further justifies this finding. As shown in the figure, the bias of the

Table 6.2: m-day ahead RMSE calculation of Kalu, Kelani, Thu Bon, Niyodo, and Monobe the datasets

Algorithm	Past Data	LSTM	GRU	RNN	LIRE	LARE	RIRE	CAS
	0-day	0.19	9.54	0.74	9.55	9.63	9.56	7.60
Kalu	1-day	48.01	45.78	48.92	57.85	57.29	57.91	55.77
	2-day	58.18	58.01	58.12	72.87	73.15	72.90	68.41
	0-day	.22	.23	.23	.27	.27	.27	.21
Kelani	1-day	0.51	0.72	0.72	0.85	0.85	0.85	0.49
	2-day	0.98	1.71	1.88	2.31	2.31	2.35	0.96
	0-day	35.78	33.80	35.09	40.12	38.33	40.19	28.19
Thu Bon	1-day	9.29	7.56	8.21	3.74	0.12	3.74	2.17
	2-day	31.90	29.11	31.44	36.05	33.80	36.35	25.61
	0-day	61.90	66.13	58.88	70.17	71.89	69.01	41.11
Niyodo	1-day	7.21	7.21	5.29	5.28	8.20	5.28	5.51
	2-day	55.78	55.80	52.45	66.76	70.19	67.23	38.96
	0-day	12.55	10.97	11.88	16.54	16.99	16.54	10.09
Monobe	1-day	1.54	0.70	0.66	0.72	1.22	0.71	.46
	2-day	11.80	10.79	11.23	10.98	11.39	10.94	9.67

Cascaded-ANFIS is zero, while other algorithm shows a significant increase or decrease in bias. NSE and KGE values of the Cascaded-ANFIS are 0.82 and 0.86, respectively. However, the Kalu River dataset does not provide the best NSE and KGE values for the Cascaded-ANFIS. Moreover, the proposed algorithm also shows an RMSE of 47.6. The GRU algorithm gives the least RMSE in this case study, having 39.54.

6.4.6 Discussion of model results and accuracy

Table 6.2 presents the overall performance of the R-R model for tested 7 algorithms (including Cascaded-ANFIS) based on RMSE for all rivers. The table further presents the RMSE values under the three scenarios that were tested. The Monobe River has its best results for scenario 2. That means the previous day's rainfall has created the runoff. However, that finding is somewhat doubtful as the Monobe River is just 71 km in length, and the catchment area is around 508.2 km^2. Therefore, one would expect scenario 1 in action for this type of river. Nevertheless, out of the 508.2 km^2 catchments, 91% is forest. Therefore, the finding of scenario 2 has its own merits for this Monobe River.

Niyodo River presents similar results. The R-R model gives better results for Scenario 2. This finding can be justified as it has a catchment area of 1560 km^2 even though the

river starts from 1982 m in MSL while having 124 km. The Thu Bon River has around 10000 km^2 catchment area, and thus scenario two can be justified for this river too. The impact of time changes on Sri Lankan rivers. Both Kelani and Kalu rivers have the same day rainfall to runoff, even though the catchment areas are larger. However, as understood the river flow paths have steep slopes and are widely famous for the steep slopes of the Kelani and Kalu rivers. In addition, downstream of these two rivers are highly urbanized and that could impact the speed of runoff.
However, these results can be further justified using hydrologic models. Nevertheless, the RMSE values among scenarios are not that much deviated. Therefore, a good scenario can be selected. The coefficient of correlation presents solid results on the R-R model developed based on the Cascaded-ANFIS algorithm. It produced the best R values for all five rivers. As stated earlier, these five river basins are in different countries and have different characteristics. Irrespective of them, the Cascaded-ANFIS algorithm outperformed all other tested algorithms. Therefore, the results can be generalized to any river basin.

6.5 Summary and conclusions

This study proposed an efficient and accurate Cascaded-ANFIS-based model for rainfall-runoff. The model was evaluated using five case studies in three countries: Japan, Vietnam, and Sri Lanka. The investigation was carried out to predict the streamflow by the influence of past data. The river case studies were selected based on the history of disasters due to heavy rainfall. Each river's dataset was examined to check the best configuration of past rainfalls affecting the streamflow volume. It was found that the Kalu River and the Kelani River located in Sri Lanka do not affect the streamflow when the past day's rainfall is considered. However, the Thu Bon River in Vietnam, the Niyodo River, and Monobe Rivers in Japan strongly connected the past day's rainfalls. This study employed six state-of-the-art regression algorithms to compare the performance of the proposed algorithm. These algorithms were selected due to their heavy use in the literature on hydrological fact predictions. The algorithms used in the study for streamflow prediction are Linear Regression, Lasso Regression, Ridge Regression, LSTM, GRU, and RNN. The case studies were evaluated using reliable hydrological parameters such as NSE, KGE, Bias, and RMSE.

The results indicated that the proposed algorithm outperforms the other algorithms in every case study except the Kalu River dataset. However, the bias calculation showed that the proposed algorithm has zero bias. Countless rivers exist with rainfall-runoff hazards in the world. Hence, the developed R-R model can be treated as a generic model for streamflow prediction for future studies.

References

[1] Bibhuti Bhusan Sahoo, Ramakar Jha, Anshuman Singh, and Deepak Kumar. Long short-term memory (lstm) recurrent neural network for low-flow hydrological time series forecasting. *Acta Geophysica*, 67(5):1471–1481, 2019.

[2] MS Horritt and PD Bates. Evaluation of 1d and 2d numerical models for predicting river flood inundation. *Journal of hydrology*, 268(1-4):87–99, 2002.

[3] Wenchao Qi, Chao Ma, Hongshi Xu, Zifan Chen, Kai Zhao, and Hao Han. A review on applications of urban flood models in flood mitigation strategies. *Natural Hazards*, 108(1):31–62, 2021.

[4] Michael W Van Liew, JG Arnold, and DD Bosch. Problems and potential of autocalibrating a hydrologic model. *Transactions of the ASAE*, 48(3):1025–1040, 2005.

[5] Jim W Hall, Shawn A Boyce, Yue-Ling Wang, Richard J Dawson, Stefano Tarantola, and Andrea Saltelli. Sensitivity analysis for hydraulic models. *Journal of Hydraulic Engineering*, 2009.

[6] Rogerio Cichota, Ana Lucia Berretta Hurtado, and Quirijn de Jong Van Lier. Spatio-temporal variability of soil water tension in a tropical soil in brazil. *Geoderma*, 133(3-4):231–243, 2006.

[7] Timo Lohmann, Amanda S Hering, and Steffen Rebennack. Spatio-temporal hydro forecasting of multireservoir inflows for hydro-thermal scheduling. *European Journal of Operational Research*, 255(1):243–258, 2016.

[8] Yukai Ding, Yuelong Zhu, Jun Feng, Pengcheng Zhang, and Zirun Cheng. Interpretable spatio-temporal attention lstm model for flood forecasting. *Neurocomputing*, 403:348–359, 2020.

[9] Dragan A Savic, Godfrey A Walters, and James W Davidson. A genetic programming approach to rainfall-runoff modelling. *Water resources management*, 13(3): 219–231, 1999.

[10] Ali Danandeh Mehr and Vahid Nourani. A pareto-optimal moving average-multigene genetic programming model for rainfall-runoff modelling. *Environmental modelling & software*, 92:239–251, 2017.

[11] PC Nayak, KP Sudheer, and SK Jain. River flow forecasting through nonlinear local approximation in a fuzzy model. *Neural Computing and Applications*, 25(7): 1951–1965, 2014.

[12] A Sezin Tokar and Peggy A Johnson. Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4(3):232–239, 1999.

[13] AR Senthil Kumar, KP Sudheer, SK Jain, and PK Agarwal. Rainfall-runoff modelling using artificial neural networks: comparison of network types. *Hydrological Processes: An International Journal*, 19(6):1277–1291, 2005.

[14] Sanaga Srinivasulu and Ashu Jain. A comparative analysis of training methods for artificial neural network rainfall–runoff models. *Applied Soft Computing*, 6(3): 295–306, 2006.

[15] Fernando Machado, Miriam Mine, Eloy Kaviski, and Heinz Fill. Monthly rainfall–runoff modelling using artificial neural networks. *Hydrological Sciences Journal–Journal des Sciences Hydrologiques*, 56(3):349–361, 2011.

[16] CL Wu and Kwok Wing Chau. Rainfall–runoff modeling using artificial neural network coupled with singular spectrum analysis. *Journal of Hydrology*, 399(3-4): 394–409, 2011.

[17] Guangyuan Kan, Jiren Li, Xingnan Zhang, Liuqian Ding, Xiaoyan He, Ke Liang, Xiaoming Jiang, Minglei Ren, Hui Li, Fan Wang, et al. A new hybrid data-driven model for event-based rainfall–runoff simulation. *Neural Computing and Applications*, 28(9):2519–2534, 2017.

[18] Tiesong Hu, Fengyan Wu, and Xiang Zhang. Rainfall–runoff modeling using principal component analysis and neural network. *Hydrology Research*, 38(3):235–248, 2007.

[19] Vahid Nourani, Mehdi Komasi, and Akira Mano. A multivariate ann-wavelet approach for rainfall–runoff modeling. *Water resources management*, 23(14):2877–2894, 2009.

[20] NJ De Vos and THM Rientjes. Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation. *Hydrology and Earth System Sciences*, 9(1/2):111–126, 2005.

[21] 23CL Wu, Kwok Wing Chau, and Yok Sheung Li. Methods to improve neural network performance in daily flows prediction. *Journal of Hydrology*, 372(1-4): 80–93, 2009.

[22] Elnaz Sharghi, Vahid Nourani, Hessam Najafi, and Amir Molajou. Emotional ann (eann) and wavelet-ann (wann) approaches for markovian and seasonal based modeling of rainfall-runoff process. *Water resources management*, 32(10):3441–3456, 2018.

[23] Taher Rajaee, Hadi Ebrahimi, and Vahid Nourani. A review of the artificial intelligence methods in groundwater level modeling. *Journal of hydrology*, 572:336–351, 2019.

[24] Vahid Nourani, Aida Hosseini Baghanam, Jan Adamowski, and Mekonnen Gebremichael. Using self-organizing maps and wavelet transforms for space–time pre-processing of satellite precipitation and runoff data in neural network based rainfall–runoff modeling. *Journal of hydrology*, 476:228–243, 2013.

[25] Niranjan Pramanik, Rabindra K Panda, and Adarsh Singh. Daily river flow forecasting using wavelet ann hybrid models. *Journal of Hydroinformatics*, 13(1):49–63, 2011.

[26] F-J Chang, Jin-Ming Liang, and Yen-Chang Chen. Flood forecasting using radial basis function neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(4):530–535, 2001.

[27] Giorgio Corani and Giorgio Guariso. Coupling fuzzy modeling and neural networks for river flood prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(3):382–390, 2005.

[28] Muhammad Shoaib, Asaad Y Shamseldin, Bruce W Melville, and Mudasser Muneer Khan. Runoff forecasting using hybrid wavelet gene expression programming (wgep) approach. *Journal of Hydrology*, 527:326–344, 2015.

[29] Vahid Nourani, Amir Molajou, Hessam Najafi, and Ali Danandeh Mehr. Emotional ann (eann): a new generation of neural networks for hydrological modeling in iot. In *Artificial intelligence in IoT*, pages 45–61. Springer, 2019.

[30] Vahid Nourani, Mehdi Komasi, and Mohammad Taghi Alami. Hybrid wavelet–genetic programming approach to optimize ann modeling of rainfall–runoff process. *Journal of Hydrologic Engineering*, 17(6):724–741, 2012.

[31] Gokmen Tayfur and Vijay P Singh. Ann and fuzzy logic models for simulating event-based rainfall-runoff. *Journal of hydraulic engineering*, 132(12):1321–1330, 2006.

[32] Yeshewatesfa Hundecha, Andras Bardossy, and HANS-WERNER WERNER. Development of a fuzzy logic-based rainfall-runoff model. *Hydrological Sciences Journal*, 46(3):363–376, 2001.

[33] Ertunga C Özelkan and Lucien Duckstein. Fuzzy conceptual rainfall–runoff models. *Journal of hydrology*, 253(1-4):41–68, 2001.

[34] Zekai Şen and Abdüsselam Altunkaynak. A comparative fuzzy logic approach to runoff coefficient and runoff estimation. *Hydrological Processes: An International Journal*, 20(9):1993–2009, 2006.

[35] Tak Kwin Chang, Amin Talei, Chai Quek, and Valentijn RN Pauwels. Rainfall-runoff modelling using a self-reliant fuzzy inference network with flexible structure. *Journal of hydrology*, 564:1179–1193, 2018.

[36] Markus Casper, Peter Gemmar, Oliver Gronz, Margret Johst, and Manfred Stueber. Fuzzy logic-based rainfall—runoff modelling using soil moisture measurements to represent system state. *Hydrological sciences journal*, 52(3):478–490, 2007.

[37] Tak Kwin Chang, Amin Talei, Sina Alaghmand, and Melanie Po-Leen Ooi. Choice of rainfall inputs for event-based rainfall-runoff modeling in a catchment with multiple rainfall stations using data-driven techniques. *Journal of Hydrology*, 545:100–108, 2017.

[38] J Eamonn Nash and Jonh V Sutcliffe. River flow forecasting through conceptual models part i—a discussion of principles. *Journal of hydrology*, 10(3):282–290, 1970.

[39] Hoshin V Gupta, Harald Kling, Koray K Yilmaz, and Guillermo F Martinez. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling. *Journal of hydrology*, 377(1-2):80–91, 2009.

[40] TN To. Strategic plan for integrated water resources management of the vu gia thu bon basin organization. *Strategic plan for integrated water resources management of the Vu Gia Thu Bon Basin Organization*, 2005.

[41] PG Cooray. Geology, with special reference to the precambrian. In *Ecology and biogeography in Sri Lanka*, pages 1–34. Springer, 1984.

[42] MMGT De Silva, SB Weerakoon, and Srikantha Herath. Modeling of event and continuous flow hydrographs with hec–hms: case study in the kelani river basin, sri lanka. *Journal of Hydrologic Engineering*, 19(4):800–806, 2014.

[43] AD Ampitiyawatta and Shenglian Guo. Precipitation trends in the kalu ganga basin in sri lanka. 2009.

Chapter 7

Investigation of computational capability of Gradient Boosting algorithms - Investigation of Malwathu river flooding, a case study in Sri Lanka.

This work is currently under review in IEEE Access (update 2022/12/21).

7.1 Introduction

Globally, floods are the most frequent natural disaster, while their impacts vary from a few households to entire regions [1]. In Sri Lanka, floods are the most frequent natural hazard, similar to the situation worldwide. 28% of the 31,063 catastrophes registered in the nation since 1974 include a flood component. Floods account for 55% of all disaster-related home damages over the same time frame (www.desinventar.net). Therefore, it is not simply a calamity that the nation experiences frequently; it also has devastating effects on the economy and communities it affects.

Since 1974, 5% of all disaster-related fatalities in the nation have been directly attributable to flooding in the island countries. It is a worrying situation when the Indian Ocean Tsunami of 2004 is taken out of the equation, and 34% of all disaster-related fatalities are attributable to flooding. Floods and other hydro-meteorological disasters

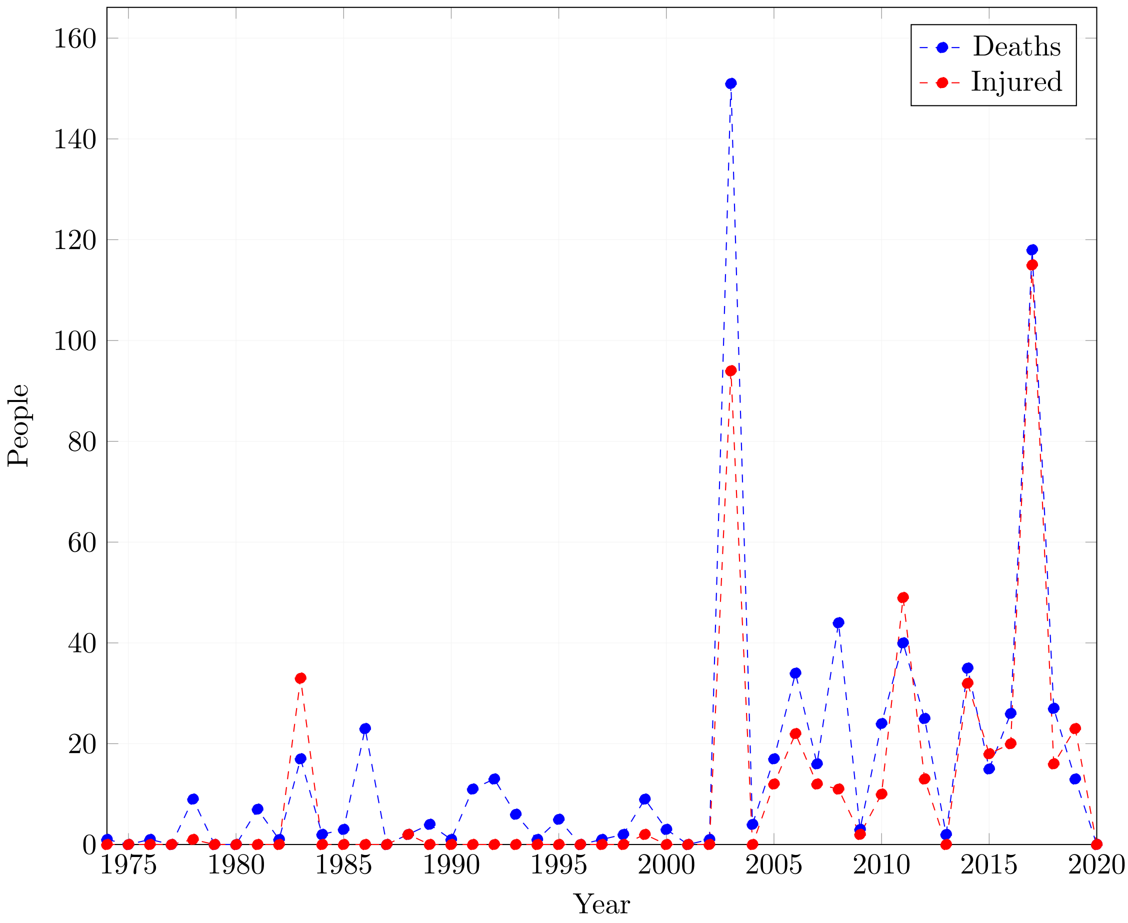


Figure 7.1: Impact of human lives due to floods during 1974-2020 in Sri Lanka

are becoming more common in Sri Lanka, according to UNDRR and ADPC [2]. The number of individuals impacted by floods is also increasing, partly due to population expansion, climate-induced rainfall unpredictability, and haphazard development activities that expose more people to flood dangers. Figure 7.1 shows past flood-related incidents and the destruction caused by them.

Global indices consistently highlight Sri Lanka's susceptibility to climate change. According to the Global Climate Risk Index, the country will be among the top 10 most vulnerable to climate change in 2018, 2019, and 2020. In terms of nations hit by climate-related disasters worldwide in 2019, the Index placed Sri Lanka second [3]. The country is now more vulnerable to flooding due to climate change-induced severe weather events and climate variability. The nation has 103 significant river basins, and 25 are particularly vulnerable to flooding (Department of Irrigation). Even though most of the nation is contained inside a river basin, rainfall anomalies have led to recurring cycles of both flood and drought, jeopardizing the country's progress in terms of development.

The World Bank estimates that the country might sustain yearly flood-related losses

and damages of up to US$380 million (Global Facility for Disaster Reduction, 2020). To assess the flood-related damages and losses and to determine the needs of the nation for recovery, the Ministries of Disaster Management, National Policies, and Economic Affairs (2016), in partnership with the European Union, UNDP, and the World Bank, conducted thorough post-disaster needs assessments (PDNA). In addition to providing an estimate of the damages, losses, and recovery requirements, these evaluations also identified several weaknesses in Sri Lanka's disaster risk management system, including the transmission of last-mile early warnings and local disaster response, among others.
The lack of catastrophe risk assessment has been a significant impediment to developing a reliable early warning system, particularly for floods. Despite several attempts, none of the flood risk assessments went beyond just charting the high-risk zones. Such efforts fell short of anticipated outcomes due to a lack of scientifically developed and practically validated methods to evaluate flood hazards and data gaps. Furthermore, the ability to execute large-scale disaster risk reductions, particularly for floods, is constrained by the lack of clarity on the components of disaster risk. Therefore, there is a higher demand for accurate and efficient river flow forecasting method investigation.

Forecasting models use previous data to make accurate predictions regardless of their structural types, such as nonlinear, linear, short, and extended memory. Previous studies' findings have shown that the results produced by the models such as Autoregressive Integrated Moving Average (ARIMA), Radial Basis Function (RBF), Adaptive Network-based Fuzzy Inference System (ANFIS), and Artificial Neural Network (ANN) were sufficiently accurate and were suitable for forecasting hydrological time series [4–9]. However, in recent studies, gradient boosting-based regression algorithms such as extreme Gradient Boosting (XGBoost) and Light Gradient Boosting (Lightgbm) showed satisfactory results in forecasting problems. Therefore, this study will consider three main river flow forecasting algorithms: Cascaded-ANFIS [10], XGBoost [11], and LightGBM [12].

For river flow forecasting, researchers have recently used a variety of methods, including radial basis function neural network (RBFNN) and RBFNN-GA [13], Emotional neural network (ENN)[14], multilayer perceptron (MLP), support vector regression (SVR), and random forest (RF) [15], a hybrid approach based on ANN and cooperation search algorithm (2021), convolution neural network (CNN) [16], Hybrid Machine Learning [17], coactive neuro-fuzzy inference system (CANFIS) [16, 18] and finally, a stochastic and neuro-fuzzy-embedded technique [19]. The absence of attention to nonlinear patterns and factors affecting the flow series is the most obvious flaw in the forecast research that has already been discussed. Regardless of the flow series' nonlinear structure, they developed models or a robust model for forecasting the flow series.

Additionally, models with comparable structures were considered while evaluating the performance of other models, and the impact of the flow series nonlinear features on models with different structures was not assessed concurrently. This effect is crucial and advantageous because, as several studies have shown [20–22], the flock series' dynamic properties directly affect the models' accuracy. Thus, the sensitivities and weaknesses of the models may be identified by taking into account the flow series' nonlinear dynamic properties, and the most effective model can be chosen by looking at these aspects. The rationale behind the current study is found here.

The results of this study provide benchmarks for estimating daily river flow and insight into physical forecasting and the impact of nonlinear time series patterns on model performance.

7.2 Methodology

This section introduces the overall methodology of the study, including prediction algorithms, the evaluation parameters of the dataset and the models.

7.2.1 Statistical evaluation criterion

The evaluation of the results of this study was done by associating the following parameters. The statistical evaluation parameters accomplished the evaluation of the models.

1. Correlation of coefficient (R)
2. Percent-Bias (bias)
3. Nash Sutcliffe model efficiency coefficient (NSE)
4. Mean Absolute Relative Error (MARE)
5. Kling-Gupta Efficiency (KGE)
6. root mean square error (RMSE)

The equations below introduce the computations of above mentioned statical evaluation parameters.

$$R = \frac{\sum(v(t) - \bar{v}(t))(u(t) - \bar{u}(t))}{\sqrt{\sum(v(t) - \bar{v}(t))^2 \sum(u(t) - \bar{u}(t))^2}} \tag{7.1}$$

$$bias = \frac{\sum_{j=1}^{k} u(t) - \bar{u}(t)}{\sum_{j=1}^{k} u(t)} \tag{7.2}$$

$$NSE = 1 - \frac{\sum_{j=1}^{k}(u(t) - \bar{u}(t))^2}{\sum_{j=1}^{k}(u(t) - \bar{v}(t))^2} \tag{7.3}$$

$$MARE = \frac{\sum_{j=1}^{k} |e_j - s_j|}{\sum_{j=1}^{k} e_j} \tag{7.4}$$

$$\begin{aligned} KGE = 1 - \sqrt{[r-1]^2 + [\alpha - 1]^2 + [\beta - 1]^2} \\ r = \frac{cov(e,s)}{\sigma(e) \cdot \sigma(s)} \\ \alpha = \frac{\sigma(s)}{\sigma(e)} \\ \beta = \frac{\mu(s)}{\mu(e)} \end{aligned} \tag{7.5}$$

$$RMSE = \sqrt{\frac{1}{q}\sum_{t=1}^{q}(u(t) - \bar{u}(t))^2} \tag{7.6}$$

where $u(t)$ is the predicted parameter, $\bar{u}(t)$ is the mean of predicted parameter$v(t)$ is the measured parameter, k is the population size and $\bar{v}(t)$ is the mean of measured parameter. The correlation coefficient (R) represents the goodness of fit. It varies from -1 to 1; the best is when it becomes 1. Bias tells the differences between predicted to measured values. The ideal bias value is 0, and 1 becomes the worst. NSE calculates the perfectness of the match between real and prediction. The results of the NSE can vary between minus infinity being the worst and 1 being the ideal [23].

7.2.2 Gradient Boosting Algorithms

Gradient boosting is a type of machine learning predictive algorithm. It is based on the suspicion that when previous models are coupled with the best possible upcoming model, the overall prediction error is minimized. Setting the desired outcomes for this subsequent model is crucial to minimizing errors. Each new model advances in the space of potential predictions for each training instance in a manner that reduces prediction error. This technique is called "gradient boosting" because target outcomes are defined for each case based on the gradient of the error about the prediction [24]. Each case's goal result will differ depending on how changing a case's forecast affects the overall prediction error.

7.2.3 Extreme Gradient Boosting Algorithm (XGBoost)

Chen et al. [25] created the XGBoost algorithm. It was developed specifically to increase computational effectiveness and model performance. In an ensemble strategy known as "boosting," adding more models fixes errors introduced by earlier models. Gradient boosting is a technique that creates new models that predict the residuals of older models combined to produce the final prediction. Gradient-boosting machines are used in a novel and expandable way that has been shown to increase the computational efficiency of boosted tree algorithms. The model addition process is repeated only when there is a noticeable improvement. A gradient descent method reduces the loss when adding new models. In 2015, XGBoost had finished 17 of the 29 ML projects that had been submitted to Kaggle. Speed was significantly boosted by using many CPU cores and reducing the look-up times of individual trees created with XGBoost. This method is constructed in R and Python using the SciKit-Learn [26] package and uses novel regularization techniques.

7.2.4 Light Gradient Boosting Algorithm (LightGBM)

The LightGBM [27] algorithm from Microsoft is an open-source GBDT. The histogram-based algorithm is the foundation for the parallel voting decision tree technique, which speeds up training, uses less memory, and integrates complex network connectivity to maximize parallel learning [28, 29]. At each iteration, the local voting choice for the top k characteristics and the global voting decision for the top 2k attributes are made. LightGBM uses the leaf-wise method to determine which leaf has the most significant splitter gain.

7.2.5 Cat Gradient Boosting Algorithm (CatBoost)

Gradient boosting is a powerful machine-learning method that can handle problems with various features, noisy data, and complex interactions. CatBoost, a machine learning method based on gradient-boosting decision trees (GBDT), was introduced by Yandex developers in 2017 [30].

Table 7.1: Machine learning algorithm parameters.

Algorithm	Parameter	Value
XGBOOST	objective	reg:linear
	colsample_bytree	0.3
	learning_rate	0.01
	max_depth	5
	Alpha	10
	n_estimators	10
CatBoost	Iterations	500
	learning_rate	0.01
	eval_metric	MultiClass
	sampling_frequency	PerTree
	penalties_coefficient	1
	max_leaves	64
	permutation_count	4
	Depth	6
LightGBM	num_leaves	31
	objective	binary
	learning_rate	0.01
	boosting_type	Dart
ANFIS	Membership_Function	Bell
	Number_of_MFs	3
	Number_of_Inputs	3
	Iterations	100

CatBoost has advantages over other GBDT algorithms: 1. The method effectively handles category features. Traditional GBDT methods can replace categorical traits with fair average label values. 2. CatBoost combines many category properties. CatBoost applies a greedy approach to integrate all categorical traits and combinations in the current tree with all categorical features in the dataset.
CatBoost can be used to address gradient bias. Each iteration of GBDT produces a weak learner, and each learner is taught using the gradient of the preceding learner. The total findings from each learner's categorization make up the output [31].

7.2.6 Model parameter tuning

This study's parameter tuning of the selected algorithm was done according to Table 10.1. Each of the algorithm parameters was introduced separately concerning the algorithm. The tuning was done by repeating the task multiple times until the best configuration of parameters was achieved.

7.2.7 Application: Malwathu Oya - Sri Lanka

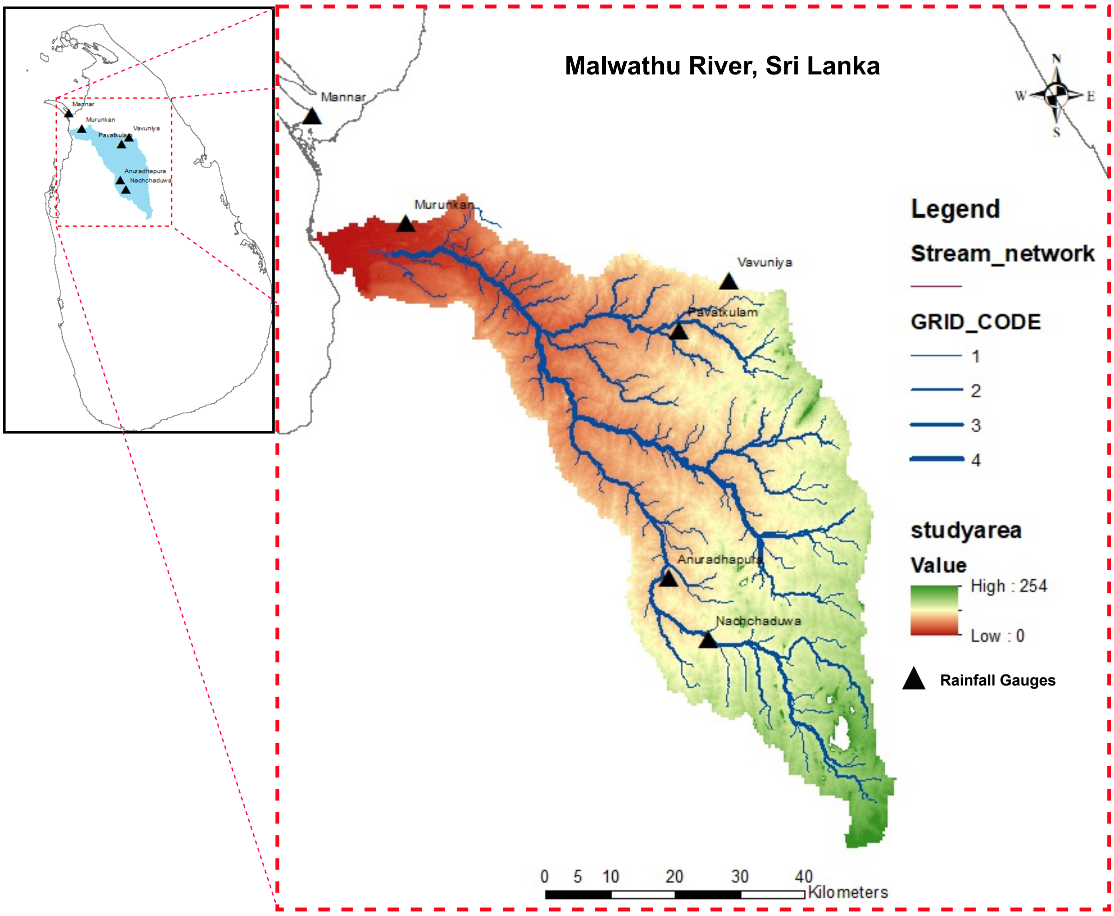


Figure 7.2: River basin of the Malwathu River in Sri Lanka

Malwathu Oya, with a length of 162 km, is the second-largest river in Sri Lanka (catchment area: 3284 km^2). It rises in the North Central Province's (766 m MSL) Ritigala Hills and empties into the sea in Arippu in the Mannar District [32]. Districts in Vavuniya and Mannar are traversed by it. In the upper catchment, a sizeable portion of the basin extends over the Anuradhapura district before narrowing extensively. There are 1,450 small reservoirs in the basin, while the upper catchment has five big reservoirs. About 410,000 people live in the basin; farmers make up most of them. The poverty headcount index is 7.6, 3.4, and 20.1, respectively, in the districts of Anuradhapura, Vavuniya, and Mannar (Department of Census and Statistics, 2012/13).

In the Malwathu Oya basin, severe flood events have been documented in 2011, 2014, and 2016. The breaching of several small reservoirs caused a significant flood event in December 2014 that is thought to be the worst since 1957, flooding several rural villages in downstream districts of Vavuniya and Mannar [33]. A functional and efficient early

Table 7.2: Descriptive analysis of the dataset of the Malwathu River - Sri Lanka

	Murrunkan	Pavattakulam	Nachchiduva	Vavniya	Mannar	Apura	Kappachichiya
Sample count	4765	4765	4765	4765	4765	4765	4765
mean	2.55	3.08	4.18	3.91	2.58	4.09	0.16
std	9.76	11.24	17.08	12.89	11.15	13.04	0.51
min	0	0	0	0	0	0	0.0025
max	161.50	185.00	417.97	225.70	350.90	192.50	6.20

warning system is required for the river basin to provide homes at risk of flooding with the time to prepare. The Department of Irrigation is conducting a basin-wide study with money from the World Bank to create a flood model and a hydro-meteorological observation system hopefully.

7.2.7.1 Dataset

The dataset used in this study was a combination of six rainfall gauges as inputs and water level station data as an output. Here, Murrunkan, Pavattakulam, Nachchiduva, Vavuniya, Mannar, and Anuradhapura (Apura) rain gauges were considered as the inputs (unit in centimetres). The Water level measurement at the Kappachichiya location was taken as the output (unit in meters). The data was collected between 2005 to 2018. The total number of samples at each input and output variable was 4765. The dataset was divided into training and testing with a 7 to 3 ratio. The overall descriptive analysis is shown in Table 7.2.

7.3 Results and Discussion

This section showcases the model predictability of the water level against rainfall. Each algorithm's performance is discussed separately, and an overall comparative analysis is presented. Here, the outputs of the gradient boosting models were considered the inputs of the ANFIS algorithm due to the curse of dimensionality issue of the ANFIS algorithm. Therefore, it can be stated that the ANFIS model provides an Ensemble structure where the gradient boosting algorithms (XGBoost, CatBoost, and LightBoost) are base algorithms, and the ANFIS is the final estimator.
The experiments were conducted with three different usages of the dataset as follows.

1. With one day before past data $(t + t_{-1})$
2. With two days before past data $(t + t_{-1} + t_{-2})$

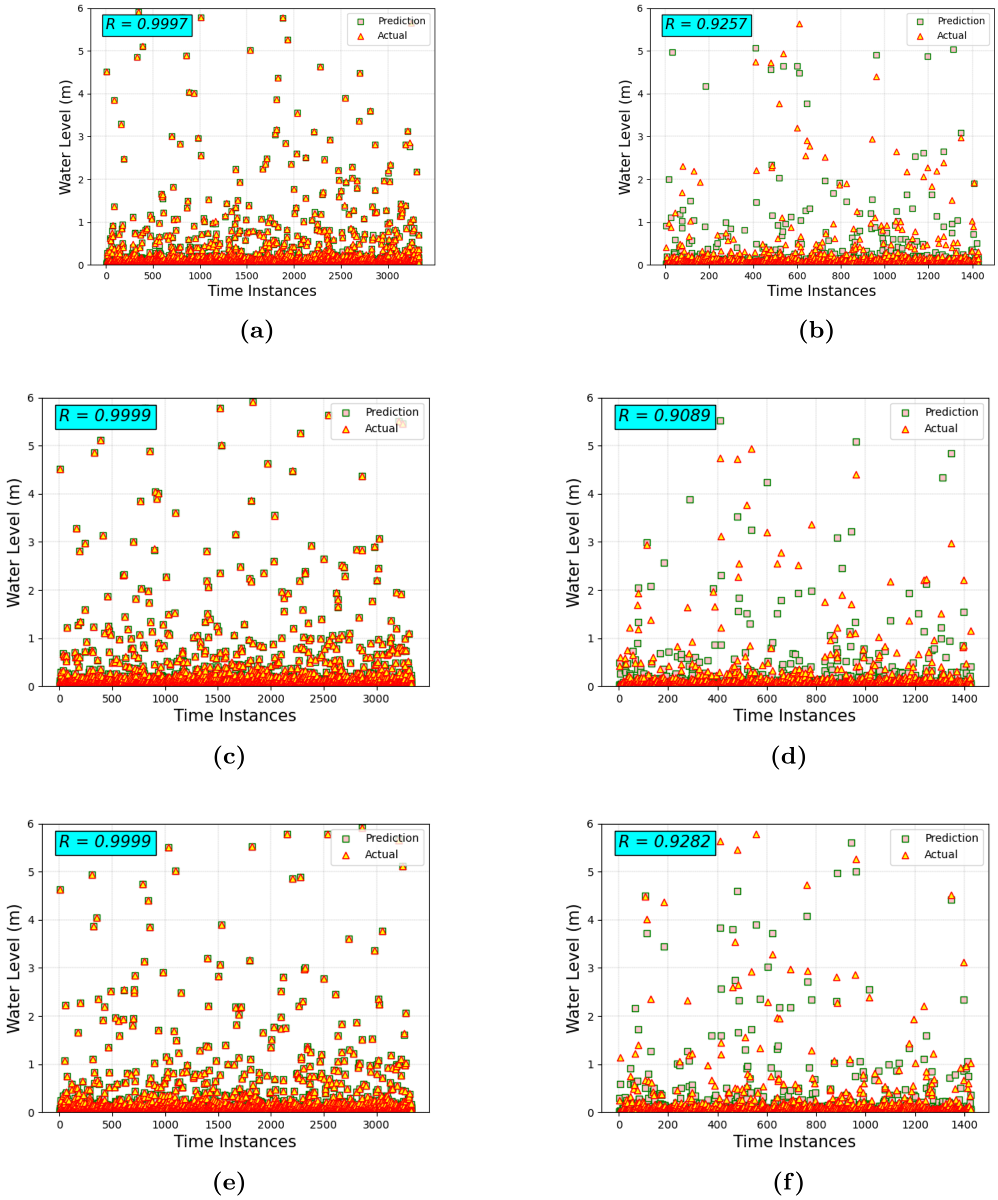


Figure 7.3: Comparission between predicted and the actual water level for water levels of Malwathu River in Sri Lanka using xgboost algorithm: (a) With one day before past data (Training) ; (b) With one day before past data (Testing); (c) With two days before past data (Training); (d) With two days before past data (Testing); (e) With three days before past data (Training); (f) With three days before past data (Testing)

3. With three days before past data $(t + t_{-1} + t_{-2} + t_{-3})$

These lag times were decided according to the civil engineering expertise of water flow with respect to the geographical structure of the case study.

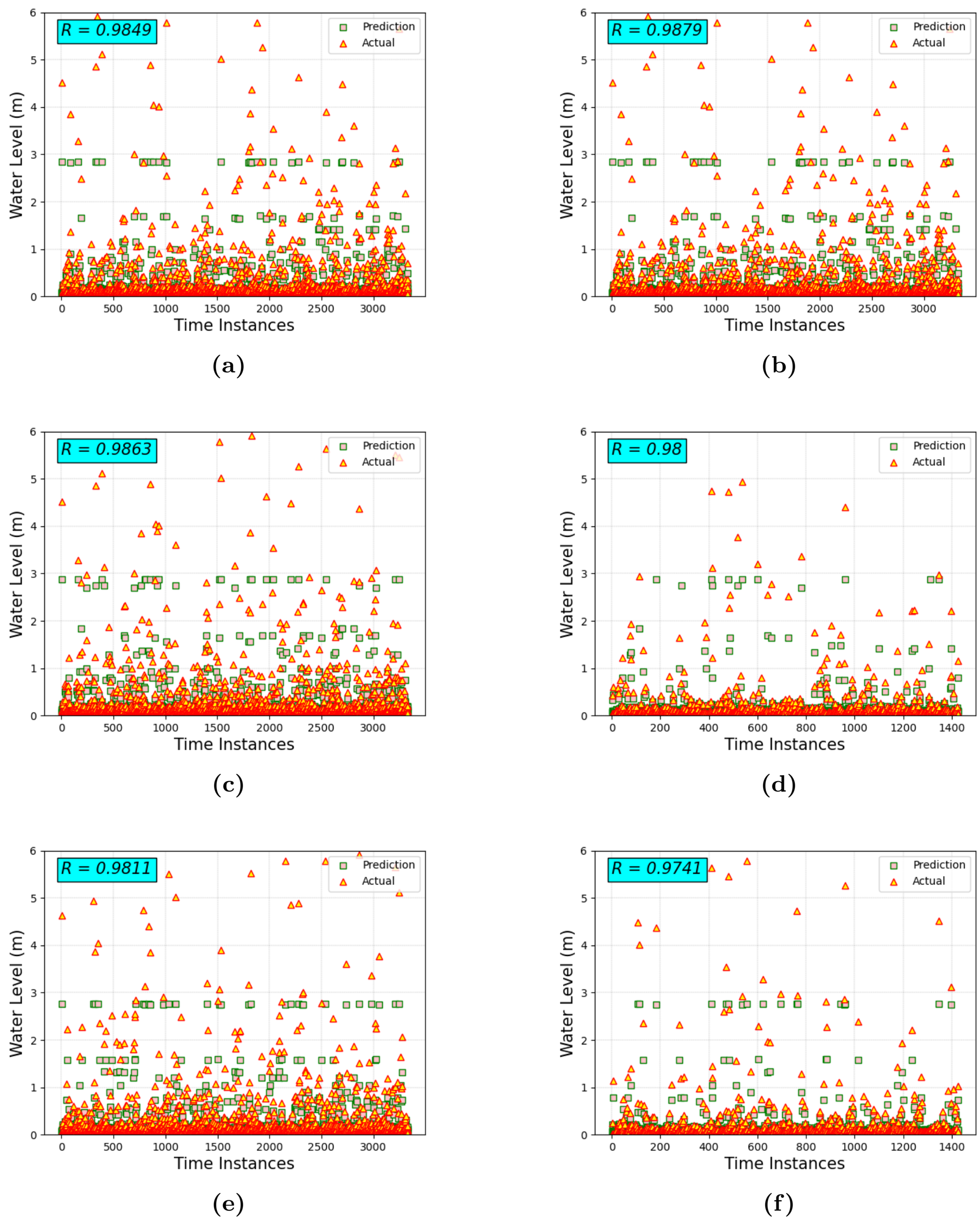
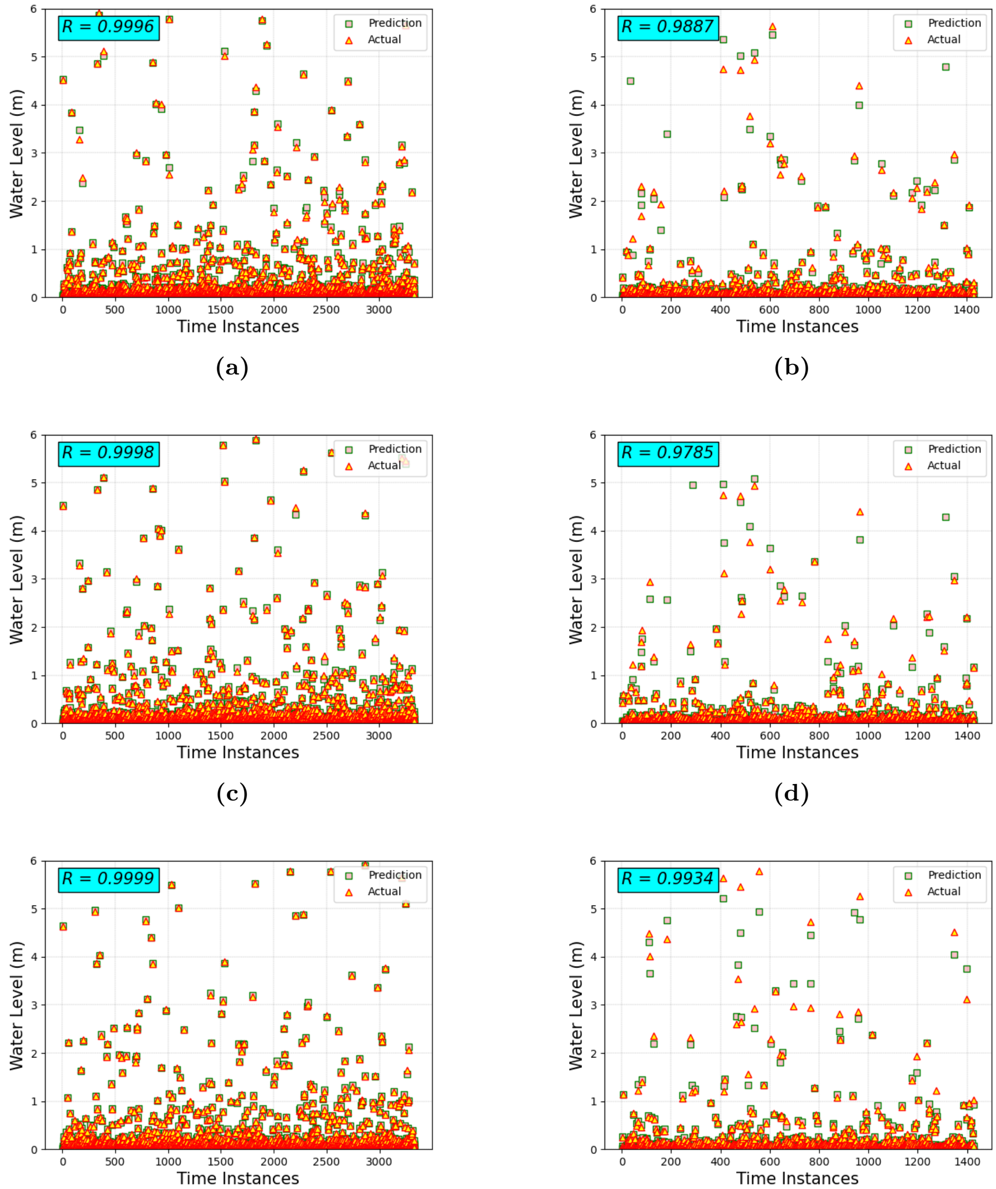


Figure 7.4: Comparission between predicted and the actual water level for water levels of Malwathu River in Sri Lanka using LightGBM algorithm: (a) With one day before past data (Training) ; (b) With one day before past data (Testing); (c) With two days before past data (Training); (d) With two days before past data (Testing); (e) With three days before past data (Training); (f) With three days before past data (Testing)

7.3.1 Extreme Gradient Boosting Algorithm (XGBoost)

The performances of the XGBoost algorithm are presented in this section. Figure 7.3 shows that the water level prediction is plotted against the time instances. As stated before, the experiments were conducted at three different lag times. The correlation coefficient (R) at each occurrence is shown in the figure. In the XGBoost model, the



(a) (b) (c) (d) (e) (f)

Figure 7.5: Comparison between predicted and the actual water level for water levels of Malwathu River in Sri Lanka using CatBoost algorithm: (a) With one day before past data (Training) ; (b) With one day before past data (Testing); (c) With two days before past data (Training); (d) With two days before past data (Testing); (e) With three days before past data (Training); (f) With three days before past data (Testing)

highest R score was given by the three days past dataset with 0.9282 for the testing. It is also noticeable that the R Score varies in an exciting way with the lag time. The highest to lowest R scores were presented as 3-day ¿ 1-day ¿ 2-day configurations.

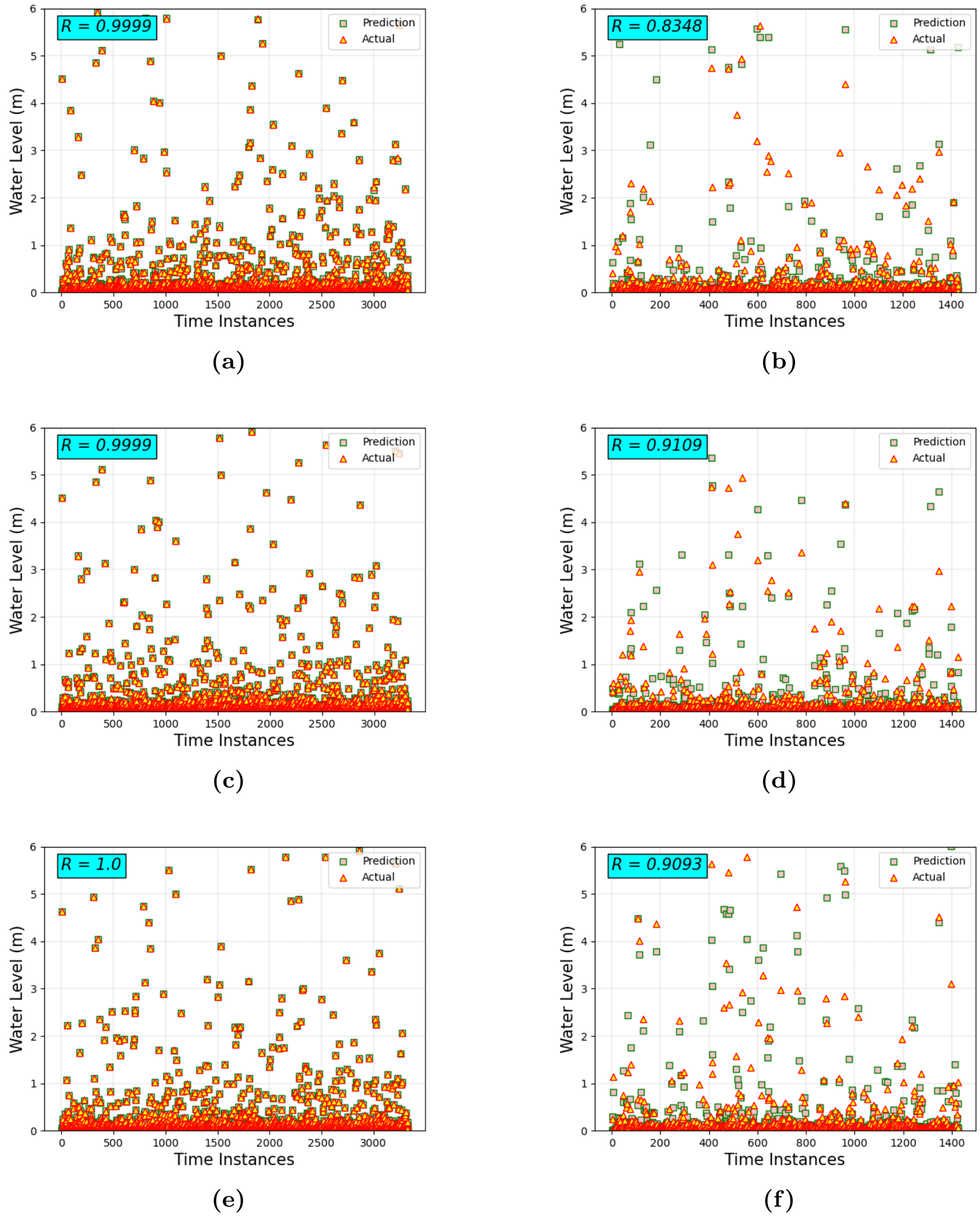


Figure 7.6: Comparison between predicted and the actual water level for water levels of Malwathu River in Sri Lanka using ensemble algorithm: (a) With one day before past data (Training) ; (b) With one day before past data (Testing); (c) With two days before past data (Training); (d) With two days before past data (Testing); (e) With three days before past data (Training); (f) With three days before past data (Testing)

7.3.2 Light Gradient Boosting Algorithm (LightGBM)

The performances of the LightGBM algorithm are presented in this section. Figure 7.4 shows that the water level prediction is plotted against the time instances. As stated before, the experiments were conducted at three different lag times. The correlation coefficient (R) at each occurrence is shown in the figure. In the LightGBM model, the highest R score was given by the 1-days past dataset with 0.9879 for the testing. It is also noticeable that the R Score varies in an exciting way with the lag time. The highest to lowest R scores were presented as 1-day ¿ 2-day ¿ 3-day configurations.

7.3.3 Cat Gradient Boosting Algorithm (CatBoost)

The performances of the CatBoost algorithm are presented in this section. Figure 7.5 shows that the water level prediction is plotted against the time instances. As stated before, the experiments were conducted at three different lag times. The correlation coefficient (R) at each occurrence is shown in the figure. In the CatBoost model, the highest R score was given by the 3-day past dataset with 0.9934 for the testing. It is also noticeable that the R Score varies in an exciting way with the lag time. The highest to lowest R scores were presented as 1-day ¿ 2-day ¿ 3-day configurations, which was the same pattern as XGBoost.

7.3.4 Adaptive Network Based Fuzzy Inference System (ANFIS) and Gradient Boosting methods as Ensemble model

The performances of the ensemble algorithm are presented in this section. Figure 7.6 shows that the water level prediction is plotted against the time instances. As stated before, the experiments were conducted at three different lag times. The correlation coefficient (R) at each occurrence is shown in the figure. In the ensemble model, the highest R score was given by the 2-day past dataset with 0.9109 for the testing. It is also noticeable that the R Score varies in an exciting way with the lag time. The highest to lowest R scores were presented as 2-day ¿ 3-day ¿ 1-day configurations.

7.3.5 Comparison with State-of-the-art Regression Model

Moreover, State-of-the-art algorithms were considered in this study to enhance the comparative analysis of the results. The following algorithms were considered state-of-the-art.

1. Grated Recurrent Unit (GRU)
2. Long Short Time Memory (LSTM)
3. Recurrent Neural Networks (RNN)
4. Lasso Regression (LASSO)

Table 7.3: State-of-the-art regression models performances evaluation

Algorithm	Configuration	Bias	MARE	RMSE	NSE	KGE
CATBOOST	1-day	2.64	0.08	0.08	0.98	0.94
	2-day	2.57	0.10	0.10	0.95	0.92
	3-day	1.60	0.08	0.07	0.98	0.95
ENSEMBLE	1-day	5.24	0.27	0.32	0.64	0.82
	2-day	-4.15	0.27	0.21	0.82	0.90
	3-day	-14.17	0.31	0.26	0.77	0.78
GRU	1-day	18.81	0.35	0.20	0.85	0.73
	2-day	6.36	0.30	0.19	0.86	0.84
	3-day	-0.31	0.31	0.19	0.86	0.88
LASSO REGRESSION	1-day	-0.16	0.45	0.20	0.85	0.89
	2-day	-0.12	0.65	0.24	0.77	0.87
	3-day	1.28	0.76	0.26	0.73	0.85
LIGHTGBM	1-day	-0.35	0.47	0.22	0.82	0.59
	2-day	-3.66	0.47	0.20	0.83	0.62
	3-day	0.53	0.45	0.23	0.82	0.62
LSTM	1-day	10.14	0.31	0.19	0.86	0.81
	2-day	14.64	0.32	0.19	0.86	0.77
	3-day	1.99	0.30	0.18	0.87	0.87
LINEAR REGRESSION	1-day	-0.16	0.42	0.19	0.85	0.89
	2-day	-0.08	0.40	0.19	0.86	0.90
	3-day	0.57	0.41	0.19	0.86	0.88
RNN	1-day	12.35	0.36	0.20	0.84	0.79
	2-day	21.63	0.36	0.21	0.83	0.70
	3-day	19.15	0.39	0.21	0.82	0.73
XGBOOST	1-day	1.31	0.35	0.19	0.86	0.91
	2-day	4.29	0.35	0.21	0.82	0.84
	3-day	0.53	0.33	0.20	0.86	0.90

5. Linear Regression (Linear)

Hyper-parameter tuning methods tuned the parameters of the above-mentioned algorithms.

Table 7.3 shows a comprehensive performance evaluation of nine different algorithms. These algorithms were used to develop a model for the Malwathu River water level prediction. As in the Table, three combinations of the dataset were used, and five statical parameters were calculated as Percent Bias (Bias), Mean Absolute Relative Error (MARE), Root Mean Square Error (RMSE), Nash-Sutcliffe Efficiency (NSE), and Kling-Gupta Efficiency (KGE).

The lowest bias value of the experiments was shown by the Linear Regression algorithm model with -0.08 at 3-day configuration, while RNN showed the highest bias value of 21.63 at 2-day configuration. However, the gradient-boosting methods showed competitive bias values of 1.60, 0.53, and 0.53 when using CatBoost, LightBoost, and XGBoost algorithms, respectively.

The MARE and RMSE values were similar for all most all the algorithms. CatBoost Algorithm scored the lowest MARE and RMSE with 0.08 and 0.07, respectively. The most significant MARE value is 0.76 for lasso regression at a 3-day configuration. The ensemble model gave the highest RMSE with 0.32 at 1-day configuration. Overall for the error-wise evaluation, the gradient boosting algorithm gives the lowest than the black-box models.

When comparing each algorithm's NSE and KGE values, the highest value is given by the CatBoost algorithm with 0.98 and 0.95, respectively.

Overall, these results convey that the CatBoost algorithm, a gradient-boosting algorithm, outperforms the other algorithms in almost all evaluation criteria. It is also noticeable that the ensemble model performance does not provide a higher state when compared with the other algorithms.

7.4 Conclusion

Simulating hydrological models is high in computational cost due to the numerous data points and input-output dimensions. Though black-box algorithms perform well in the litreture for prediction and forecasting, the excessive use of computational resources is challenging to handle. Therefore, this study was conducted to evaluate and analyse the computational performances of the gradient-boosting algorithms in hydrological prediction and forecasting. CatBoost, extreme boost and light gradient boost algorithms were considered gradient-boosting algorithms due to their vast popularity in the scientific community.

This study focuses on a specific case study called Malwathu River, located in Sri Lanka. There have been vast amounts of human and infrastructure damage due to this river's sudden flooding. Therefore, this work utilizes the rainfall and water level dataset of the Malwathu River to train hydrological models to predict and forecast the river's flooding. The data was collected from the year 2005 to 2018 with six rainfall gauges in the river basin of the Malwathu River. Moreover, three configurations were established to do the experiments with 1-day, 2-day, and 3-day lag times.

The results were evaluated under six statical evaluation criteria; namely, correlation of coefficient (R), Precent-Bias (bias), Nash Sutcliffe Model efficiency coefficient (NSE), Mean Absolute Relative Error (MARE), Kling-Gupta Efficiency (KGE), and Root mean square error (RMSE). These evaluation criteria are well-reputed for hydrological model analysis in the literature. Moreover, an Adaptive network-based fuzzy Inference system (ANFIS) based ensemble model was generated to check if the performance can be enhanced by using gradient boosting algorithms as base models.

The results of this study show that the CatBoost hydrological model outperforms other algorithms. The results were compared with nine different algorithms, including black-box algorithms (LSTM, GRU, RNN). This study concludes that the Cat gradient boosting algorithm can predict the hydrological modelling better than the general black-box algorithms. As for the future aspect, implementing a real-time flood warning system in Sri Lanka using a model that consumes less power and computational cost can be introduced.

References

[1] Johan Askman, Olof Nilsson, and Per Becker. Why people live in flood-prone areas in akuressa, sri lanka. *International Journal of Disaster Risk Science*, 9(2):143–156, 2018.

[2] Chadia Wannous and German Velasquez. United nations office for disaster risk reduction (unisdr)—unisdr's contribution to science and technology for disaster risk reduction and the role of the international consortium on landslides (icl). In *Workshop on World Landslide Forum*, pages 109–115. Springer, 2017.

[3] Daniela Baum. Global climate risk index 2019-who suffers most from extreme weather events? weather-related loss events in 2017 and 1998 to 2017. 2018.

[4] Mohini P Darji, Vipul K Dabhi, and Harshadkumar B Prajapati. Rainfall forecasting using neural network: A survey. In *2015 international conference on advances in computer engineering and applications*, pages 706–713. IEEE, 2015.

[5] Stephen Davies, Arthur Gueneau, Dawit Mekonnen, Claudia Ringler, and Sherman Robinson. Irrigation and water management in the indus basin: infrastructure and

management strategies to improve agricultural productivity. *Agriculture and the rural economy in Pakistan: issues, outlooks, and policy priorities. University of Pennsylvania Press, Philadelphia*, pages 117–170, 2016.

[6] Ata Allah Nadiri, Keyvan Naderi, Rahman Khatibi, and Maryam Gharekhani. Modelling groundwater level variations by learning from multiple models using fuzzy logic. *Hydrological sciences journal*, 64(2):210–226, 2019.

[7] Fariba Niromandfard, Abbas KhasheiSiuki, and Ali Shahidi. Evaluation of the neuro-fuzzy and hybrid wavelet-neural models efficiency in river flow forecasting (case study: Mohmmad abad watershed). *Journal of Watershed Management Research*, 10(19):211–221, 2019.

[8] Mustafa Al-Mukhtar. Random forest, support vector machine, and neural networks to modelling suspended sediment in tigris river-baghdad. *Environmental monitoring and assessment*, 191(11):1–12, 2019.

[9] Hadi Sanikhani, Ozgur Kisi, Eisa Maroufpoor, and Zaher Mundher Yaseen. Temperature-based modeling of reference evapotranspiration using several artificial intelligence models: application of different modeling scenarios. *Theoretical and applied climatology*, 135(1):449–462, 2019.

[10] Namal Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. A novel optimization algorithm: Cascaded adaptive neuro-fuzzy inference system. *International Journal of Fuzzy Systems*, 23(7):1955–1971, 2021.

[11] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[13] Haitham Abdulmohsin Afan, Mohammed Falah Allawi, Amr El-Shafie, Zaher Mundher Yaseen, Ali Najah Ahmed, Marlinda Abdul Malek, Suhana Binti Koting, Sinan Q Salih, Wan Hanna Melini Wan Mohtar, Sai Hin Lai, et al. Input attributes optimization using the feasibility of genetic nature inspired algorithm: Application of river flow forecasting. *Scientific Reports*, 10(1):1–15, 2020.

[14] Zaher Mundher Yaseen, Sujay Raghavendra Naganna, Zulfaqar Sa'adi, Pijush Samui, Mohammad Ali Ghorbani, Sinan Q Salih, and Shamsuddin Shahid. Hourly river flow forecasting: application of emotional neural network versus multiple machine learning paradigms. *Water Resources Management*, 34(3):1075–1091, 2020.

[15] Dostdar Hussain and Aftab Ahmed Khan. Machine learning techniques for monthly river flow forecasting of hunza river, pakistan. *Earth Science Informatics*, 13(3): 939–949, 2020.

[16] Chao Huang, Jing Zhang, Longpeng Cao, Long Wang, Xiong Luo, Jenq-Haur Wang, and Alain Bensoussan. Robust forecasting of river-flow based on convolutional neural network. *IEEE Transactions on Sustainable Computing*, 5(4):594–600, 2020.

[17] Hai Tao, Ali Omran Al-Sulttani, Ameen Mohammed Salih Ameen, Zainab Hasan Ali, Nadhir Al-Ansari, Sinan Q Salih, and Reham R Mostafa. Training and testing data division influence on hybrid machine learning model process: application of river flow forecasting. *Complexity*, 2020, 2020.

[18] Joseph Tripura, Parthajit Roy, and AK Barbhuiya. Simultaneous streamflow forecasting based on hybridized neuro-fuzzy method for a river system. *Neural Computing and Applications*, 33(8):3221–3233, 2021.

[19] Khadije Lotfi, Hossein Bonakdari, Isa Ebtehaj, Mohammad Rezaie-Balf, Pijush Samui, Ahmed A Sattar, and Bahram Gharabaghi. River flow forecasting using stochastic and neuro-fuzzy-embedded technique: a comprehensive preprocessing-based assessment. In *Water Engineering Modeling and Mathematic Tools*, pages 519–549. Elsevier, 2021.

[20] Farhang Rahmani and Mohammad Hadi Fattahi. Investigation of denoising effects on forecasting models by statistical and nonlinear dynamic analysis. *Journal of Water and Climate Change*, 12(5):1614–1630, 2021.

[21] Mohammad Hadi Fattahi, Naser Talebbeydokhti, GR Rakhshandehroo, Abolfazl Shamsai, Ehsan Nikooee, et al. Fractal assessment of wavelet based techniques for improving the predictions of the artificial neural network. *Journal of Food, Agriculture & Environment*, 9(1):719–724, 2011.

[22] Zhong-kai Feng and Wen-jing Niu. Hybrid artificial neural network and cooperation search algorithm for nonlinear river flow time series forecasting in humid and semi-humid regions. *Knowledge-Based Systems*, 211:106580, 2021.

[23] J Eamonn Nash and Jonh V Sutcliffe. River flow forecasting through conceptual models part i—a discussion of principles. *Journal of hydrology*, 10(3):282–290, 1970.

[24] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3):1937–1967, 2021.

[25] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[26] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[27] Qi Meng, Guolin Ke, Taifeng Wang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, and Tie-Yan Liu. A communication-efficient parallel algorithm for decision tree. *Advances in Neural Information Processing Systems*, 29, 2016.

[28] Sanjay Ranka and Vineet Singh. Clouds: A decision tree classifier for large datasets. In *Proceedings of the 4th knowledge discovery and data mining conference*, volume 2, 1998.

[29] Ruoming Jin and Gagan Agrawal. Communication and memory efficient parallel decision tree construction. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 119–129. SIAM, 2003.

[30] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[31] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[32] T Thiruvarudchelvan and SS Sivakumar. Operational policy of the reservoirs in malwathu oya river basin to minimize flood damages in anuradhapura, vavuniya and mannar districts in northern sri lanka. 2019.

[33] TJ Meegastenna. Assessment of climate change impacts and adaptation measures to malwatu oya river basin in north central province of sri lanka. In *International Workshop on Innovation of Developing the Strategy for Impact Assessment of and Adaptation to the Climate Change as the "New Normal"(CLIMATE). In Proceedings of the 3rd World Irrigation Forum, Bali, Indonesia*, pages 1–7, 2019.

Chapter 8

Rice seeds classification based on its age.

This work is presented at the International Conference of Machine Vision (ICMV 2022) in Rome.

8.1 Introduction

Over 2000 years, rice has remained an important cultural component and the staple food in Japan, with 3.5 million rice producers, and each Japanese consumes 70 kilograms of rice on average annually [1]. There are around 300 rice varieties that can be found all around the country, among approximately 40,000 different kinds of rice varieties that exist globally.

Especially with the increased scarcity of resources such as water and land has led to a trend of maintaining sustainable rice production through technological improvements. The decline or stagnancy of yield levels caused by low grain quality and increase in production costs due to high dependence on agricultural inputs is something that researchers are passionate about improving further. These constraints go hand in hand since planting rice seeds that are less capable of producing the maximum possible yield degenerates the maximum capacity of scarce resources in rice cultivation.

However, despite these constraints, rice production should increase drastically over the next generation to satisfy the global food demand, particularly for the poor. Therefore, the significance of producing more rice with a reduced /controlled supply of resources is an alarming challenge to secure the food supply and the social, economic, and water sustainability of the Asian region, which is firmly attached to rice culture [2]. Previous studies have identified the age of rice seeds after harvesting as a crucial factor determining the quality of rice seeds primarily used for rice cultivation. Therefore, this research is conducted to develop a system to identify or confirm rice seeds' word-of-mouth age more efficiently, which ultimately could serve as one component for determining the overall quality of rice seeds.

Japan is highly vulnerable to natural catastrophes such as earthquakes, hurricanes, and flooding due to its climate and geography. Therefore, crop failure or harvest loss can occur due to adverse weather conditions. Hence, there are scenarios in that age seeds tend to be used for cultivation in Japan. The attention required for treatment could be taken with rice seeds by identifying the age of rice seeds with more reliability. Three wide Japanese rice varieties harvested in 2012, 2016, and 2020 were used in this research. The rice varieties used in this study are Akitakomachi, Koshihikari, and Yangdao-8. According to our knowledge, there is no rice seed dataset based on the harvested age in the scientific community. Therefore, this research mainly contributes in two ways: implementing a novel rice seed dataset based on age variation and proposing a novel approach for classifying the dataset using SURF-BOF-based Cascaded ANFIS. This manuscript is presented as follows. Section 2 investigates the related works. The methodology of this study is discussed in section 3. Section 4 is dedicated to results and discussion, and the conclusion is presented as the final section.

8.2 Related Works

There is very little literature on the relationship between germination and the rice seed age. Nevertheless, to our knowledge, no studies have focused on rice seed classification based on age. The following paragraphs introduce the related works on the relationship between germination and rice seed age but are not limited only to rice. Furthermore, the related studies on using Speeded-Up Robust Features (SURF), Bag-of-Features (BOF), and Cascaded Adaptive Network-based Fuzzy Inference System (ANFIS) in image classification are presented. The relationship between germination and the rice seed age was extensively studied by Jones et al. in 1926 [3]. They state that the seed age is inversely proportional to the germination success rate for most rice varieties. They have used eight rice varieties in the research and researched six years of age gaps. A study shows that germination against the wheat seed age has the same relationship as in the earlier study. They state that the germination rate decrease with the rate of 0.243% h^{-1} of ageing [4]. Yun et al. have researched canola seeds to evaluate the germination rate due to seed ageing [5]. They have also concluded that the germination rate of older is lower than that of newer.

Furthermore, Tabatabaei in 2014 [6] and Ibrahim et al. in 2013 [7] also provided research outcomes of two types of research conducted based on seed germination against ageing and concluded the same results as mentioned above. These indications provide substantial importance in seed classification based on age-wise. Wu and Tsai introduced a noise reduction of leaf images. They have used background removal and the ROI extraction methods as novel implementations and succeeded in achieving 92.13% accuracy [8]. Moreover, an object recognition technology has been introduced using python and MNIST dataset modification by Karayaneva and Hintea. They have used five machine learning algorithms, including neural networks and achieved 87%-98% accuracy on object recognition [9]. A combination of KNN and ANN is used for the dragon fruit classification tasks. The study showed that the machine learning method performed six times better than manual classification [10].

In past studies on rice grain classification, Tzu-Yi proposed a model using a sparse-representation-based 30 varieties classification of rice grains [11]. In the paper [12], the authors have introduced a model using BOF-SVM to classify and recognize vegetable pests. They have contributed a rice dataset of 30 types by using microscope images of the rice grains. They have obtained an overall accuracy of 89.1% as a result. They have considered a total of four classes in the dataset. The feature extraction of the research was performed using the scale-invariant feature transform (SIFT) descriptor, and they scored an accuracy of 91.56% for the classification. Furthermore, the SIFT-BOF-based image recognition system is developed by Yuki et al. [13], and they propose a fuzzy code book to reduce computational complexity.

Furthermore, Raj Kumar et al. [14] used the SURF feature descriptor to generate a dataset of 1000 samples to identify Fungal Blast disease in rice seeds. They have tried several Machine Learning (ML) algorithms for the classification, and the best results were obtained using Convolutional Neural Network (CNN) based VGG-16 algorithm. The experiment shows an overall accuracy of 71.28%. Moreover, using the Jointly Multi-Model Bag of Feature algorithm, research has been conducted to classify the soybean's quality utilizing computer vision techniques [15]. They have used the SURF feature descriptor and the spatial layout of "L $\times$ a $\times$ b $\times$ colour features" to extract features from the dataset. They used the Low-Rank Representation technique to reduce the feature dimension, and SVM was used as the classification algorithm. The experimental results show 95.6% of accuracy as the classification results.

Furthermore, the BOF-based Almond classification system was introduced by Abozar et al. [16]. They have constructed a novel dataset for Almond seeds' sweet and bitter classification. They used the SIFT feature descriptor as the feature extractor and tested three classification algorithms based on Support Vector Machines (SVM): K-NN based on SVM, L-SVM, and Chi-SVM. The researchers concluded that the Chi-SVM outperforms the other two algorithms with an accuracy of 91%.

However, past studies on age-wise classification do not exist according to our knowledge. Therefore, the research gap in age-wise classification is explicit.

8.3 Methodology

8.3.1 Dataset Construction

The dataset is constructed using the rice seed samples of three main rice varieties in Japan: Akitakomachi, Koshihikari, and Yangdao-8. These rice types are further separated based on the year of harvesting. The overall label set of the constructed rice dataset is shown in Table 1.

Since the research combines two classifications: the rice variety's type and the year of harvest, the labelling was completed in two manners. First, the samples are labelled according to the rice variety, and the second is the year of harvesting. Finally, the dataset is divided into three segments for training, testing, and validation. The training, testing, and validation proportions are 60%, 30%, and 10% of the total dataset. However, it is worth noting that the number of samples in each rice variety differs. The sample image data set can be illustrated in Figure 8.1.

Furthermore, according to the age of the harvested rice seeds, there are three categories: 2012, 2016, and 2020; however, as shown in Table 8.1, the Yangdao-8 consists only of two types such as 2012 and 2020, due to the unavailability of the 2016 rice seeds. The novel dataset is publicly available in the Kaggle dataset repository and named "Japanese Rice Seeds Age-wise Classification" [17].

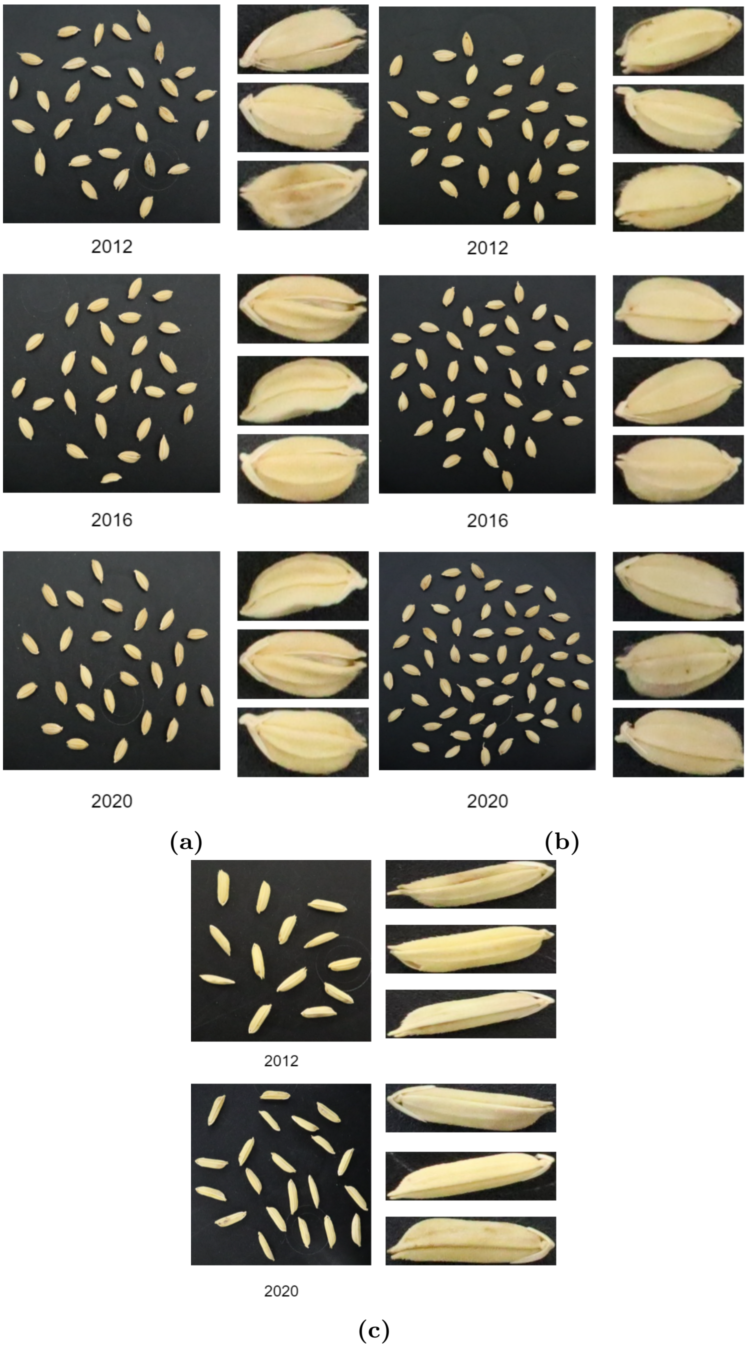


Figure 8.1: Rice seeds; a) Akitakomachi, b) Koshihikari, c) Yangdao 8

8.3.2 Feature Extraction

Features are one of the significant components in designing a classifier. This research attempts to classify similar images but rich in different surface textures. Therefore, it is crucial to investigate a feature descriptor that suits the problem and can provide rich features for the classification. SURF feature descriptor is a method that can deal with this kind of situation. The following paragraph introduces a brief description of the

Table 8.1: Structure of the Rice Seeds dataset

Rice Variety	Sample Size			Total Samples
	2012	2016	2020	
Akitakomachi	427	368	392	1187
Koshihikari	377	368	392	1319
Yangdao-8	275	N/A	235	510

SURF feature descriptor.

The SIFT and SURF algorithms follow the same principle in general. However, SURF is a three-step process: detecting interest points, local neighbourhood description, and matchmaking. SURF employs square-shaped filters as a rough approximation of Gaussian smoothing. Filtering the image with a square is quicker when the integral image is used. Depending on the Hessian matrix (HS), SURF utilizes a blob detector to determine locations of interest. The determinant of the HS is utilized to assess local inconstancy around the point where this determinant is maximum [18].

In SURF, scale spaces are created using box filters of various dimensions. As a result, the scale space is examined by improving the filter dimensions rather than repeatedly lowering the image dimensions. The subsequent layers are created by gradually increasing the size of the masks utilized to filter the image.

A descriptor's purpose stands to offer a distinctive and reliable description of the image, such as the intensity allocation of the pixels. Therefore, most descriptors are generated locally, resulting in a description for each site of interest already specified.

It is necessary to determine the orientation of the interest points to provide rotational invariance. Therefore, Harr wavelets were calculated at each interest point, and vectors were generated for a specified radius of the interesting point. The most extended vector defines the point of interest's direction.

The next feature extraction step is creating the BOF [19]. Therefore, the features mentioned above are used to develop the visual vocabulary. The nearest neighbour matching is used in this process for the clustering. Then the clusters are mapped as frequency histograms of BOF. Then the histograms are used with the Cascaded-ANFIS algorithm for the classification.

8.3.3 Modified Cascaded ANFIS algorithm

According to the problem, the Cascaded-ANFIS algorithm can be utilized conveniently. Due to many feature points in this research, the Cascaded-ANFIS algorithm is used

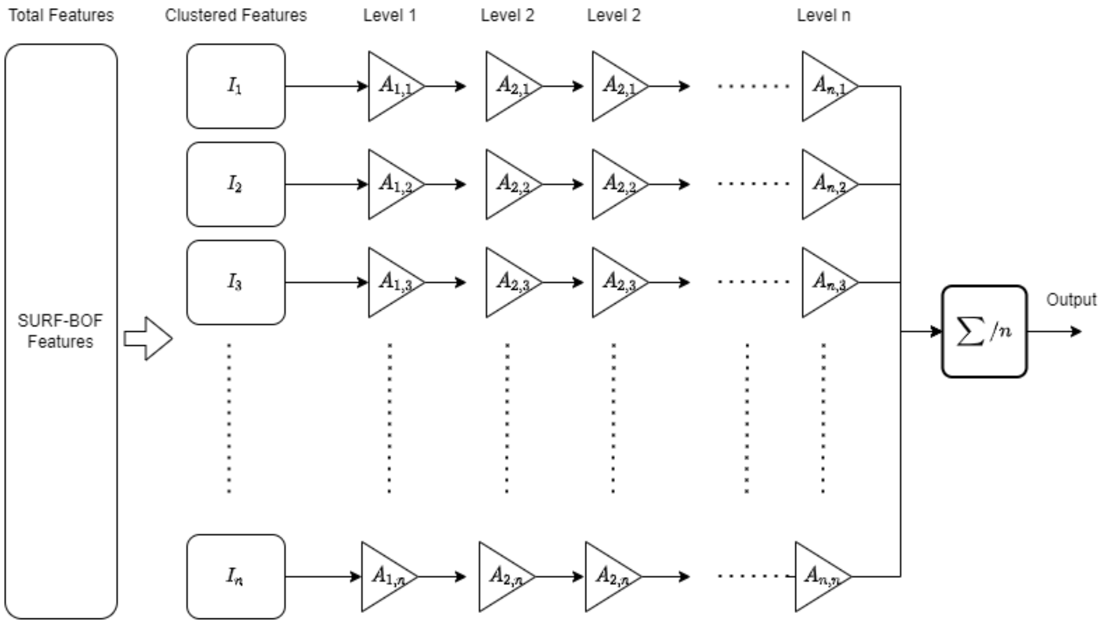


Figure 8.2: Modified Structure of the Cascaded-ANFIS algorithm.

as two different methods. As shown in Figure 8.2, the feature points were selected based on the pair selection method. However, the features were not used as pairs but as a group. For a group, ten features were assigned. Then using a 10-input 1-output ANFIS structure, the first-level outputs were generated. After that, using two-input one-output ANFIS structures, the remaining levels were operated. The pair selection method selected the best pair of inputs to propagate through the network at each level. The overall operation of the modified Cascaded-ANFIS structure is shown in Figure 8.2. A total of 500 features were extracted as the SURF-BOF features, and 50 clusters were assigned for the initial level of the Cascaded ANFIS. As in Figure 8.2, Ai,j is the ANFIS structure where i is the number of levels, and j is the number of structures. At the end of 20 levels in the modified Cascaded-ANFIS model, the total outputs were averaged to calculate the final out.

8.3.4 Experiment Platform

The experiments were carried out using a Windows 10 personal computer. The computer's processor is an Intel core i9 with 3.70 GHz, and the memory size is 64 GB. The experiments were carried out without using any GPU processing.

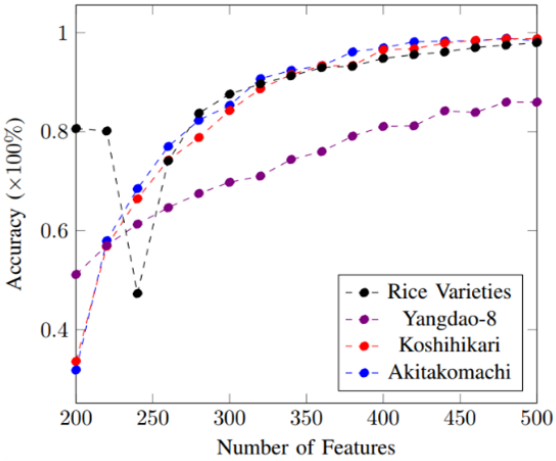


Figure 8.3: Accuracy of the classification vs number of features used to train the Cascaded-ANFIS model

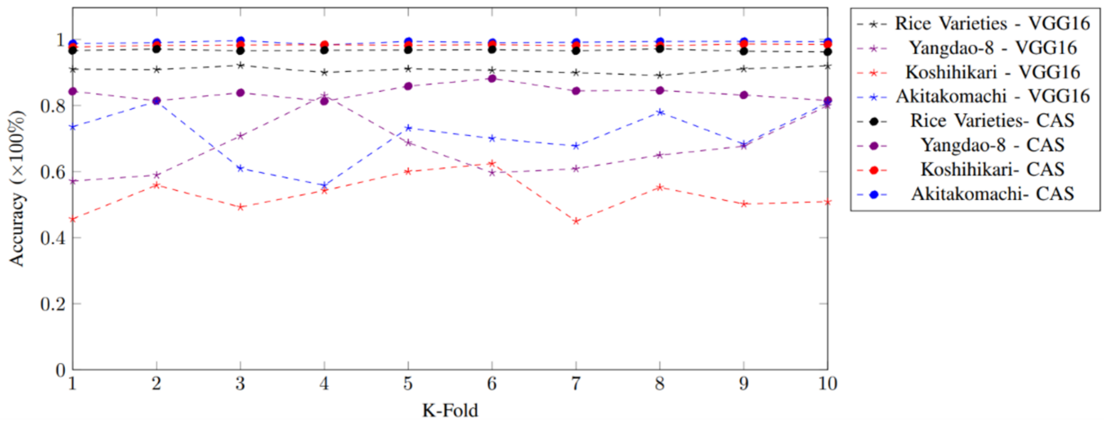


Figure 8.4: 10-Fold Cross-Validation of the Cascaded-ANFIS and VGG16 models training

8.4 Results and Discussion

The experiment results were analyzed in several aspects. The first experiment was designed to check the performance based on the number of the input feature vector. As mentioned in the above sections, the data set contains three different rice varieties. As shown in Figure 8.3, the accuracy of each dataset was calculated using the Cascaded-ANFIS algorithm with the variation of feature points starting at 200 to 500. It is clear that the around 500 feature points, the accuracy becomes stable at 98%.

A K-Fold Cross-Validation was accomplished because of the smaller size of the dataset. As shown in Figure 8.4, ten-fold cross-validation was utilized. The VGG16 model is trained and tested to compare the proposed algorithm performance with 10-fold cross-validation. The learning rate of the VGG16 model is 0.0001, and 100 iterations were used for each piece of training. The "imagenet" was taken as the weights for the VGG16

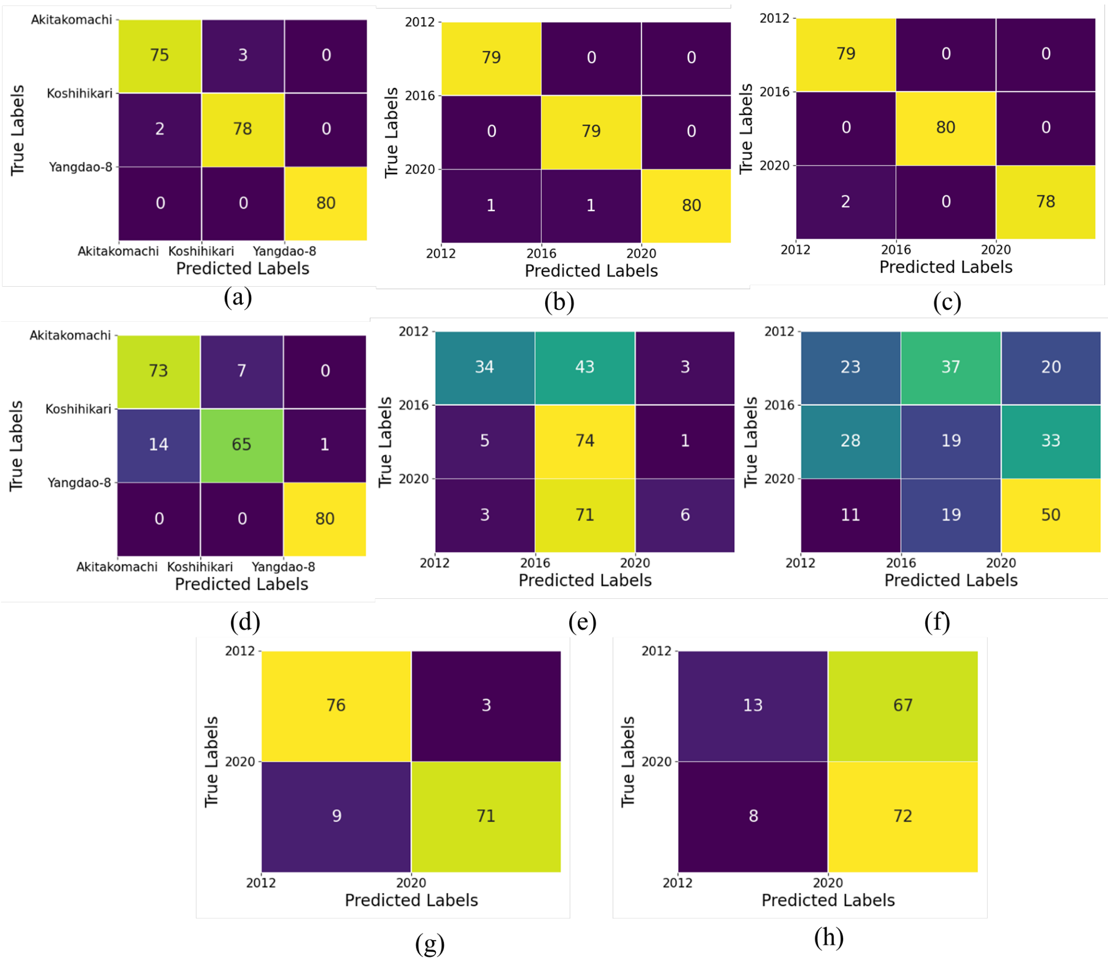


Figure 8.5: Confusion Matrix of Rice Seeds Classifications. (a) Rice Varieties: Cascaded-ANFIS, (b) Akitakomachi Age-wise: Cascaded-ANFIS, (c) Koshihikari Age-wise: Cascaded-ANFIS, (d) Rice Varieties: VGG16, (e) Akitakomachi Age-wise: VGG16, (f) Koshihikari Age-wise: VGG16, (g) Yangdao-8 Age-wise: Cascaded-ANFIS, and (h) Yangdao-8 Age-wise: VGG16.

training. The parameters used to create the Cascaded-ANFIS algorithm are as follows. The Fuzzy Inference System (FIS) of ANFIS was generated using 3-Gaussian membership functions for each corresponding ANFIS module in the Cascaded-ANFIS. The total number of input features was taken as 500, and the total number of iterations was considered 20 for this experiment.

The results showed that the Cascaded-ANFIS model performs better than the VGG16 model. As shown in Figure 8.4, the Yangdao-8 rice shows a comparatively smaller accuracy percentage. The other three classifications show better accuracy, around 95%. However, the proposed model average accuracies of Akitakomachi, Koshihikari, Yangdao-8, and Rice variety classifications are 99.15%, 98.27%, 83.86%, and 96.71%, respectively. Furthermore, confusion matrixes were generated for each experiment, as shown in Figure

Table 8.2: Evaluation parameter scores for the Cascaded-ANFIS and VGG16 classifications.

Parameter	Precision		Recall		F1-Score		Accuracy	
Algorithm	CAS	VGG16	CAS	VGG16	CAS	VGG16	CAS	VGG16
Rice Varieties	0.97	0.89	0.97	0.85	0.97	0.87	0.97	0.91
Akitakomachi Age-wise	0.99	0.67	0.99	0.68	0.99	0.97	0.99	0.56
Koshihikari Age-wise	0.99	0.45	0.99	0.90	0.99	0.60	0.99	0.45
Yangdao-8 Age-wise	0.92	0.71	0.92	0.75	0.92	0.71	0.92	0.69

8.5. Here, (a), (b), (c), and (g) sub-figures represent the proposed algorithm and (d), (e), (f), and (h) represents the VGG16 model performances. The confusion matrix is not a measure for evaluating a model but gives information about the predictions. It is required to apprehend the confusion matrix to understand other classification metrics such as average accuracy, precision, recall, and F1-Score. Each of these metrics conveys valuable information about the performance of the classification when the problem is multi-class [20]. As shown in the confusion matrixes and Table 8.2, the research was an evident success. The confusion matrix of the classification of rice varieties showed that the Yangdao-8 seeds offer 100% accuracy. This was a result of the shape of the rice seed. It was observable that the Yangdao-8 seeds are longer than the other two varieties considered in this research. However, the Akitakomachi and Koshihikari showed promising results with few false predictions.

Though the Yongdao-8 Rice has 100% accuracy in classifying varieties, the age classification showed lesser results when compared with the other two types. Furthermore, the confusion matrix of the age of Akitakomachi rice seed classification showed the best accuracy, with 99.00%. Nevertheless, the Koshihikari age classification also provided satisfactory results, with 99.00%.

8.5 Conclusion

The age of the rice seed is a vital factor for successful germination. Therefore, this study implemented an automatic rice seed age-wise classification using the SURF-BOF-based modified Cascaded-ANFIS algorithm. Furthermore, this study also contributes a novel Japanese rice seed dataset for age classification. The dataset combines three main rice varieties in Japan, and each rice variety is separated according to the year of harvestings, such as 2012, 2016, and 2020. However, the Yongdao-8 rice variety does not contribute to three different years but only 2012 and 2020. The performance evaluation was done by comparing the results with the VGG16 deep learning model. 10-Fold cross-validation was performed for each classification, and the mean accuracies demonstrate the algorithm's robustness. The classification results were analyzed by generating the confusion matrix and evaluated using precision, recall, and F1-Score. The results are promising, with an accuracy of 99% for the Koshihikari rice age classification. The classification between the rice varieties provides an accuracy of 97%. Due to the shape difference of Yandao-8 rice seeds, the individual accuracy of the seed classification from others is 100%. Investigating the relationship between the germination success rate and the rice seed quality assessment using the SURF-BOF-based Cascaded-ANFIS algorithm can be considered a future objective.

References

[1] Susumu Yamaji and Shôichi Itô. The political economy of rice in japan. In *Japanese and American Agriculture*, pages 349–365. Routledge, 2019.

[2] Tejendra Chapagain and Eiji Yamaji. The effects of irrigation method, age of seedling and spacing on crop performance, productivity and water-wise rice production in japan. *Paddy and Water Environment*, 8(1):81–90, 2010.

[3] Jenkin William Jones. Germination of rice seed as affected by temperature, fungicides, and age. *Agronomy journal*, 1926.

[4] E Soltani, S Galeshi, B Kamkar, and F Akramghaderi. The effect of seed aging on the seedling growth as affected by. *Research Journal of Environmental Sciences*, 3 (2):184–192, 2009.

[5] Yun Ren, Jianfang Zhu, Nazim Hussain, Shanlin Ma, Genru Ye, Dongqing Zhang, and Shuijin Hua. Seedling age and quality upon transplanting affect seed yield of canola (brassica napus l.). *Canadian Journal of Plant Science*, 94(8):1461–1469, 2014.

[6] SA Tabatabaei et al. The changes of germination characteristics and enzyme activity of barley seeds under accelerated aging. *Cercet. Agron. Mold*, 48:62–67, 2015.

[7] H Ibrahim, JA Oladiran, and H Mohammed. Effects of seed dormancy level and storage container on seed longevity and seedling vigour of jute mallow (corchorus olitorius). *African Journal of Agricultural Research*, 8(16):1370–1374, 2013.

[8] Yen-Ju Wu, Chun-Ming Tsai, and Frank Shih. Improving leaf classification rate via background removal and roi extraction. *Journal of Image and Graphics*, 4(2): 93–98, 2016.

[9] Yordanka Karayaneva and Diana Hintea. Object recognition in python and mnist dataset modification and recognition with five machine learning classifiers. *Journal of Image and Graphics*, 6(1):10–20, 2018.

[10] Nguyen Minh Trieu and Nguyen Truong Thinh. Quality classification of dragon fruits based on external performance using a convolutional neural network. *Applied Sciences*, 11(22):10558, 2021.

[11] Tzu-Yi Kuo, Chia-Lin Chung, Szu-Yu Chen, Heng-An Lin, and Yan-Fu Kuo. Identifying rice grains using image analysis and sparse-representation-based classification. *Computers and Electronics in Agriculture*, 127:716–725, 2016.

[12] Deqin Xiao, Jianzhao Feng, Tanyu Lin, Chunhua Pang, and Yaowen Ye. Classification and recognition scheme for vegetable pests based on the bof-svm model. *International Journal of Agricultural and Biological Engineering*, 11(3):190–196, 2018.

[13] Yuki Shinomiya and Yukinobu Hoshino. A feature encoding based on low space complexity codebook called fuzzy codebook for image recognition. *International Journal of Fuzzy Systems*, 21(1):274–280, 2019.

[14] Raj Kumar, Gulsher Baloch, Abdul Baseer Buriro, Junaid Bhatti, et al. Fungal blast disease detection in rice seed using machine learning. *International Journal of Advanced Computer Science and Applications*, 12(2), 2021.

[15] Ping Lin, Li Xiaoli, Du Li, Shanchao Jiang, Zhiyong Zou, Qun Lu, and Yongming Chen. Rapidly and exactly determining postharvest dry soybean seed quality based on machine vision technology. *Scientific Reports*, 9(1):1–11, 2019.

[16] Abozar Nasirahmadi and Seyed-Hassan Miraei Ashtiani. Bag-of-feature model for sweet and bitter almond classification. *Biosystems engineering*, 156:51–60, 2017.

[17] N. Rathnayake. Japanese rice seeds agewise classification. 2022.

[18] RMNB Rathnayake and L Seneviratne. An efficient approach towards image stitching in aerial images. In *2018 3rd International Conference on Information Technology Research (ICITR)*, pages 1–6. IEEE, 2018.

[19] Toru Tamaki, Junki Yoshimuta, Misato Kawakami, Bisser Raytchev, Kazufumi Kaneda, Shigeto Yoshida, Yoshito Takemura, Keiichi Onji, Rie Miyaki, and Shinji Tanaka. Computer-aided colorectal tumor classification in nbi endoscopy using local features. *Medical image analysis*, 17(1):78–100, 2013.

[20] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

Chapter 9

Designing and Simulation of ANFIS-based UAV controller

This work is presented at IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2021) [1].

9.1 Introduction

The quad-copter controller design is a trending topic of today's world researchers, because, the possibilities and capabilities of the quadcopter provide high expectations. The quad-copters are usually used in the fields of agriculture [2, 3], areal transportation [4, 5], mapping and exploration, fire and rescue [6, 7] and photography [8, 9].

Quad-copter has four motors, while the degrees of freedom are six. Hence, it is an underactuated mechanical system. Due to the under-actuation of the quad-copter, designing a better control algorithm is an extensive challenge. Recently there have been several quad-copter control algorithms which provide better performance in stabilization when compared with the early-stage controllers. PID control is one of the most used algorithms in quad-copter controls [10–12]. Most researchers tend to use the PID control algorithm as the initial step of developing the novel algorithms [13–15]. Some of these can be listed as follows. Neural network (NN) based controllers [16], Neural-PID based controllers [17], sliding mode controllers [17], Linear quadratic controllers [18] and ANFIS based controllers [19, 20].

Among these control algorithms, it is a proven fact that the ANFIS-based controllers provide better efficiency and stability, according to the following literature. The authors in [21] have developed a quad-copter using the ANFIS algorithm to control the attitude and altitude. The authors have compared the results between ANFIS-PD, FUZZY-PD and PD controllers. The results show that the ANFIS-based controller has better accuracy in attitude and altitude control over the other two algorithms. In [22], a Permanent Magnet Brushless DC motor drive system is implemented using ANFIS and PI controllers. The results show that the ANFIS provides better results in current and torque control over the fixed PI controller. An altitude tracking system for a drone is implemented in [23]. They have used a hybrid PD-ANFIS-based algorithm for controlling. The results were obtained for classical PID, P-D, ANFIS and Intelligent PD-ANFIS controllers. Though all the considered controllers were able to stabilize the drone, the PD-ANFIS gives faster stability as in the results section of the paper.

The authors in [24] have presented an algorithm for drone control using ANFIS and the Improved Particle Swarm Optimization algorithm (ANFIS-IPSO). The results were compared between PID, ANFIS and the proposed algorithm. The trajectory tracking error has almost come to a zero when using the ANFIS-IPSO algorithm. In [19], a trajectory control system for a quad-rotor is designed using the PSO-ANFIS algorithm. Two simulation tests have been carried out for performance analysis, such as trajectory tracking and mass loading and unloading. They compare the results with PID, ANFIS and PSO-ANFIS algorithms, proving that PSO-ANFIS provides better stability in both simulations. Hence, it is possible to do a performance analysis among state-of-the-art ANFIS-based algorithms.

The main goal of this paper is to investigate the performance of the GA-ANFIS and PSO-ANFIS in quad-copter control.

The rest of the paper is organized as follows. Section II gives a brief introduction of the related theories such as quadcopter model dynamics, PID controllers, ANFIS, GA and PSO. Then the methodology of the controller implementation is presented in section III. In section IV, the results were analysed, and finally, the conclusion and future works are presented in section V.

9.2 Related Theories

9.2.1 Quad-copter model dynamics

It is necessary fact to understand the basic dynamics of the quad-copter design before implementing a control algorithm. As mentioned in the above section, a quad-copter is a system which is under-actuated. Because there are six degrees of freedom in the quad-copter, but it has four rotors to control the movements. The six degrees of freedom can be divided into two as, translation movements (along the x-axis, y-axis and z-axis) and rotational angles (φ - roll, θ - pitch and ψ - yaw). This section is used to give a brief introduction to the quad-copter model dynamics.

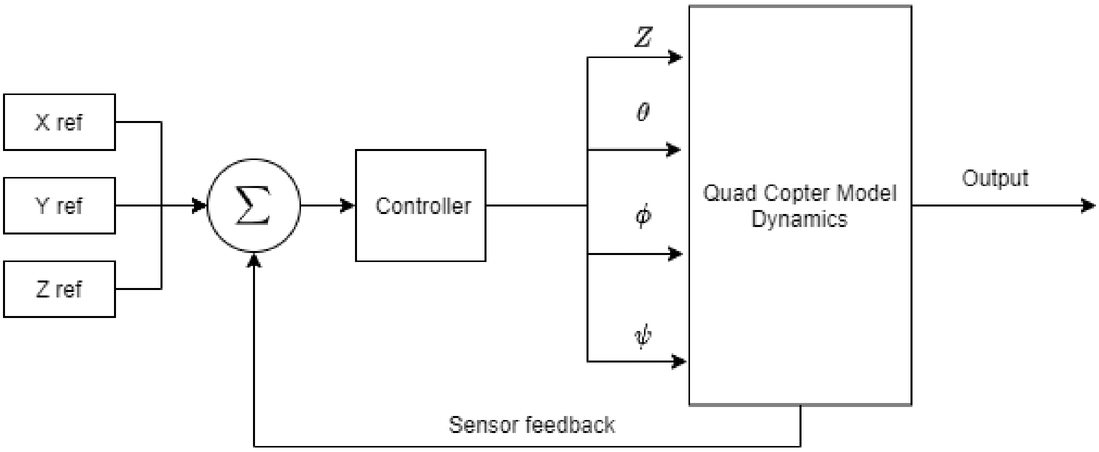


Figure 9.1: Quad-Copter Controller

Since there are two types of freedoms, kinematics can be divided into two sections as well. As referenced in [19], the translation kinematics (9.1) and rotational kinematics (9.2) can be presented as the following equations.

$$\dot{\xi} = R.V_b \tag{9.1}$$

$$\dot{\eta} = R_r.\mho_b \tag{9.2}$$

Here absolute linear velocity vector is presented as $\dot{\xi}$ and it can be expressed as $\begin{bmatrix}\dot{x} & \dot{y} & \dot{z}\end{bmatrix}$. Linear velocity is expressed using V_b and it can be presented as $\begin{bmatrix}U & V & W\end{bmatrix}^T \in \mathbb{R}^3$. R can be given as:

$$R_r = \begin{bmatrix} 1 & s_\varphi t_\theta & c_\varphi t_\theta \\ 0 & c_\varphi & -s_\varphi \\ 0 & \frac{s_\varphi}{c_\theta} & \frac{c_\varphi}{c_\theta} \end{bmatrix} \tag{9.3}$$

where Euler angle derivatives can be presented as $t_\theta = \tan\theta$, $\dot{\eta} = \begin{bmatrix} \dot{\varphi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \in \mathbb{R}^T$ As in [25], the rotor forces can be expressed as in (9.4) and the model dynamics can be represented using (9.5).

$$f_i = b.{\omega_i}^2 \tag{9.4}$$

$$\begin{cases} \dot{x} = u \\ \dot{y} = v \\ \dot{z} = w \\ \dot{u} = (c_\varphi s_\theta c_\psi + s_\varphi s_\psi)\frac{T}{m} \\ \dot{v} = (c_\varphi s_\theta c_\psi - s_\varphi c_\psi)\frac{T}{m} \\ \dot{w} = (c_\varphi c_\theta)\frac{T}{m} - g \\ \dot{\varphi} = p + s_\varphi t_\theta q + c_\varphi t_\theta r \\ \dot{\theta} = c_\varphi q - s_\varphi r \\ \dot{\psi} = \frac{s_\varphi}{c_\theta} q + \frac{c_\varphi}{c_\theta} r \\ \dot{p} = \frac{\tau_x}{j_x} + \frac{j_y - j_z}{j_x} qr \\ \dot{p} = \frac{\tau_y}{j_y} + \frac{j_z - j_x}{j_y} pr \\ \dot{r} = \frac{\tau_z}{j_z} + \frac{j_x - j_y}{j_z} pq \end{cases} \tag{9.5}$$

In (9.4), rotor thrust is denoted as b and angular velocities are ω. The inertia matrix of the quadcopter is denoted as j and j_x, j_y and j_z are the moments of inertia in (9.5).

9.2.2 Proportional Integral and Derivative (PID) control

PID control is the most commonly used control technology in control engineering. This section aims to give a brief introduction to the PID controllers. It is said that PID controllers are the workhorse of modern process control systems [26]. PID controllers are made with three components such as proportional, integral and derivative. The basic structure of the PID controller in a system is shown in Fig. 9.2. As shown in the figure, the output is controlled by three parameters (P, I, and D). Generally, PID controllers are closed-loop controllers. Therefore, the error $e(t)$ is calculated using the feedback

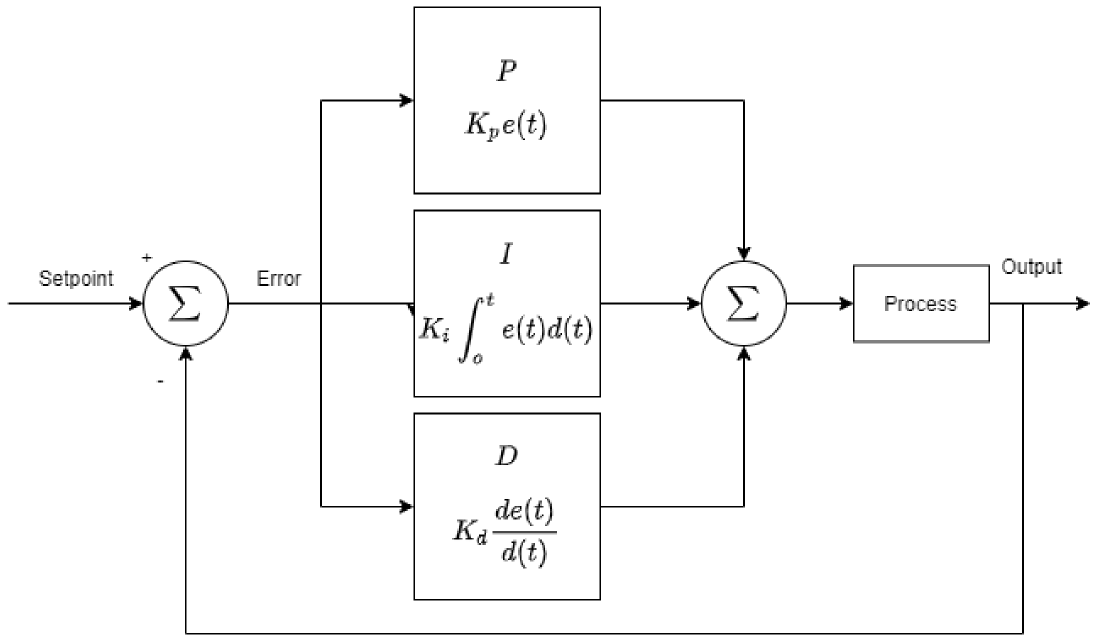


Figure 9.2: Classical PID Controller

path and the setpoint to control the system. In the figure, K_p, K_i and K_d are the gains of proportional, integral and derivative respectively.
The tuning of these gains is challenging. Hence, many researchers have proposed several types of PID tuning methods. Some of them are; the trial and error Method [27], Zeigler-Nichols Method [28], Cohen-Coon method [29], Tyreus-Luyben method [30] and autotune method. Moreover, there are some software-based tuning methods such as MATLAB PID tuning [31].

9.2.3 ANFIS

In 1978, L.A. Zadeh [32] introduced Fuzzy Logic Controllers (FLC). Since then, it has evolved to a significant level. The recent evolution of FLC is ANFIS. ANFIS is based on two well-known algorithms such as FLC and ANN. Jang in 1993 [33] proposed ANFIS as a machine learning algorithm. In this section, a brief introduction to ANFIS is presented. Fig. 9.3 shows the general structure of the ANFIS algorithm. ANFIS possess all the advantages of the ANN and the FLC [34]. Generally, ANFIS is a five-layer structure as shown in the figure. Each layer is explained in brief below.

- Layer 1: The first layer is called the fuzzification layer. Here, the memberships are generated for each input using the standard membership functions as in (9.6).

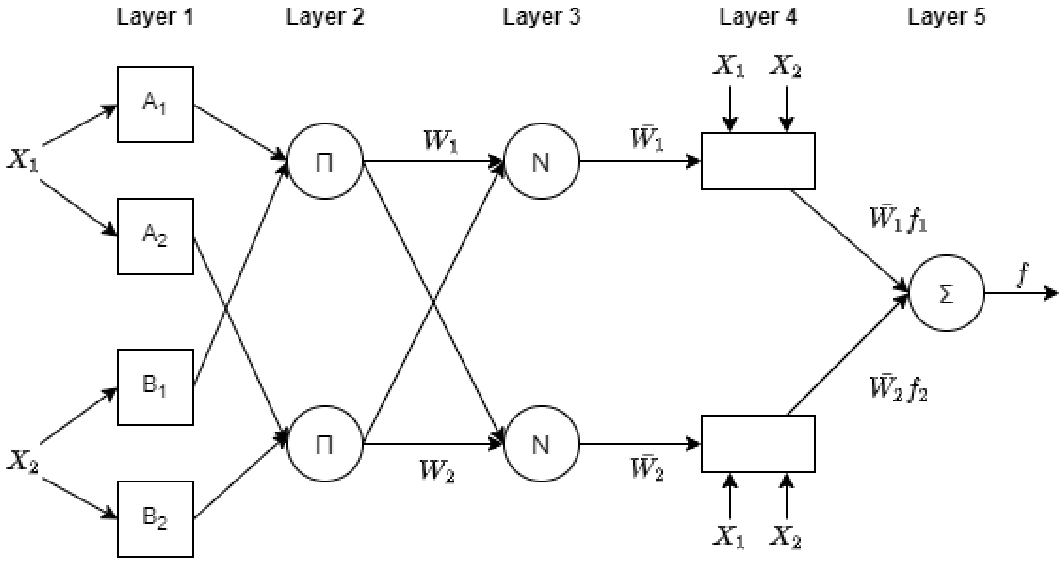


Figure 9.3: ANFIS Architecture

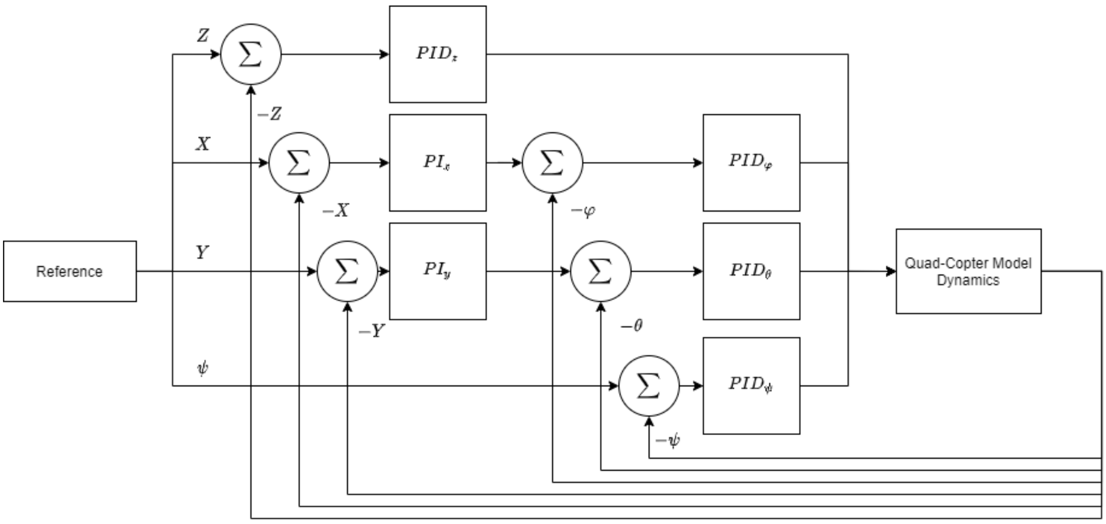


Figure 9.4: PID control of Quad-Copter

$$O_{1,i} = \begin{cases} \mu_{Ai}(X_1) \\ \mu_{Bi}(X_2) \end{cases} \tag{9.6}$$

where, linguistic labels are denoted as A_i and B_i while node function is denoted as i. There are three commonly used membership functions such as triangular (9.7), trapezoidal (9.8) and gaussian (9.9).

$$\mu_{Ai} = max(min(\frac{x - a_i}{b_i - a_i}, \frac{c_i - x}{c_i - b_i}), 0), i = 1, 2 \tag{9.7}$$

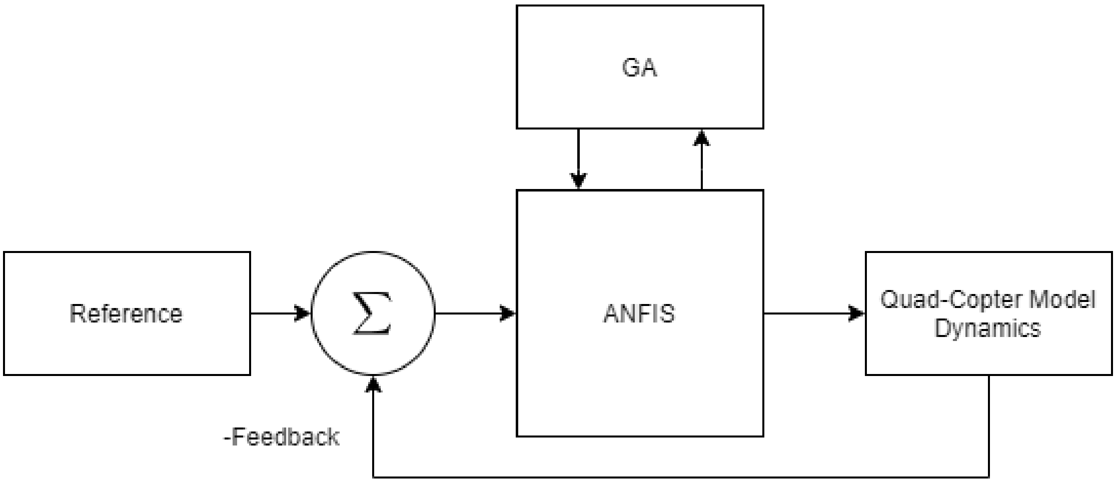


Figure 9.5: GA-ANFIS control of Quad-Copter

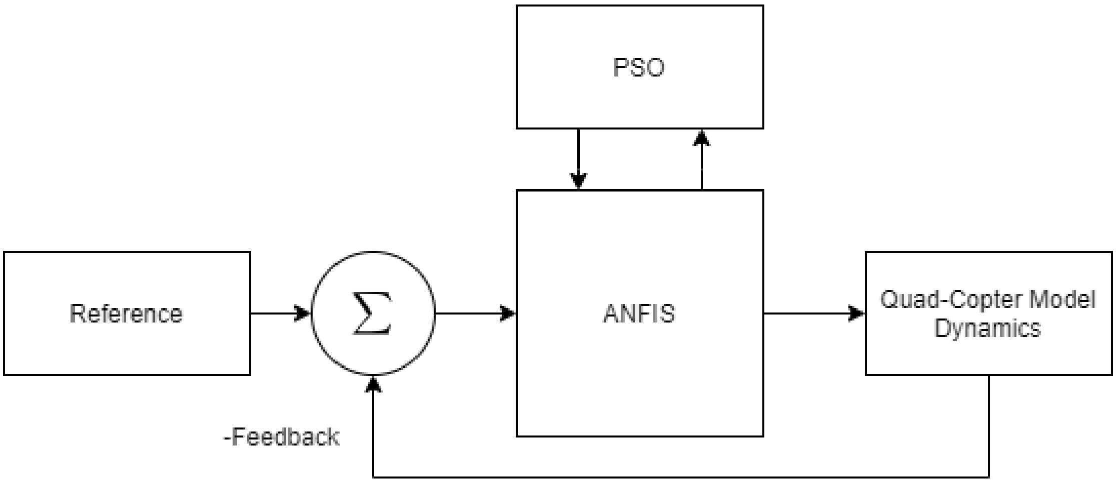


Figure 9.6: PSO-ANFIS control of Quad-Copter

$$\mu_{Ai} = max(min(\frac{x - a_i}{b_i - a_i}, 1, \frac{d_i - x}{d_i - c_i}), 0), i = 1, 2 \tag{9.8}$$

$$\mu_{Ai} = exp(-\frac{(x - c_i)^2}{\sigma_1^2}), i = 1, 2 \tag{9.9}$$

Here, a_i, b_i, c_i, d_i and σ_i are premise parameters.

- Layer 2: This later is called the rule layer. As in (9.10), the firing strength w of the each rule is generated by multiplying $\mu_{Ai}(x_1)$ and $\mu_{Bi}(x_2)$.

$$O_{2,j} = w_j = \mu_{Ai}(x_1) \times \mu_{Bi}(x_2) \tag{9.10}$$

where j is the number of the rule.

- Layer 3: This is the normalization layer. The main objective of this latter is to generate the third output by normalization by firing the strength of each corresponding rule. Normalization can be expressed as in (9.11).

$$O_{3,j} = \bar{w_j} = \frac{w_j}{\sum w_i}, j = 1, 2, ..., n \tag{9.11}$$

- Layer 4: This layer is responsible for the defuzzification. Consequent parameters (p_j, q_j, r_j) are used here as in (9.12) to do the defuzzification of the previously generated normalized firing strengths $\bar{w_j}$.

$$O_{4,j} = \bar{w_j} f_j = \bar{w_j}(p_j x_1 + q_j x_2 + r_j) \tag{9.12}$$

- Layer 5: This is the output layer. The Sum of all incoming inputs gives the overall output as in (9.13).

$$O_{5,1} = \sum_{j=1}^{n} \bar{w_j} f_j \tag{9.13}$$

9.2.4 Fitness Function

The relative importance of a design can be expressed using a fitness function. Determining the best fitness function for a certain system is an assuring way to collect better results. The most common way to describe a fitness function is using a cost function as shown in (9.14)[35].

$$F_i = (1 + \varepsilon) f_{max} - f_i \tag{9.14}$$

where, the cost function is denoted as f_i for the ith design, f_{max} is max cost and ε is used to eliminate the difficulty of F_i being zero. Considering the quadcopter control, the Integral of Time multiplied by Absolute Error (ITAE) is the most used method in determining a fitness function. The adjusting time and the overshoot are indicated by ITAE. These characteristics reflect the accuracy and the rapidity of the control system. It is scientifically proven that (9.15) gives the performance index of a quadcopter control [36].

$$J_{ITAE} = \int_0^\infty t.\,|e(t)|\,dt \tag{9.15}$$

9.2.5 Genetic Algorithms (GA)

The inspirations of Genetic Algorithms (GA) are natural selection and biological processes. There are several versions of GA in the literature. This section gives a brief introduction to GA and its construction.

GA is an optimization algorithm which tries to find the optimum solution in a solution space by iterations. First, the search for the best solution starts with a randomly generated population in the solution space. In order to continue with the iterations, GA proceed with three operators; namely, selection, crossover and mutation [37].

The principle of "Survival of the Fittest" imitates the first operation "Selection". Considering the current design set which is also called the population, this process is predisposed towards the better fit members. In (9.16), this process is expressed in mathematically[35].

$$P_i = \frac{F_i}{Q}; Q = \sum_{j=1}^{N_p} F_j \tag{9.16}$$

Here, F_i is the fitness value. N_p is the number of members in the population and P_i is the probability of the selection.

The second operator "Crossover" is about mating in biological populations. There are several crossover methods for completing this operation such as single-point crossover, two-point and k-point crossover, and uniform crossover. Crossover helps to sustain the good solutions and remove the bad solutions and move on with the new population to the next generation. Diversity in characteristics of the population is done by the last operator "Mutation". Mutation helps to escape from the local minima and propagate to the global minima.

Generally, GA encodes the design into bits of ones and zeros. Hence, GA is well known for discrete designs because of this behaviour of the algorithm.

9.2.6 Particle Swarm Optimization (PSO)

This section is used to describe the PSO algorithm in brief. PSO is first introduced in 1997 [38] by the inspiration of birds flocking. PSO is also an optimization algorithm which tries to move in the solution space to find the best solution possible called global minima. PSO also operates in three steps, namely, positions and velocity generations for each particle, updating the velocities and updating the positions. PSO also uses iterations to complete the search. The first step of the PSO is to generate a random population with random positions and velocities as in (9.17) and (9.18).

$$\boldsymbol{x}_0^i = \boldsymbol{x}_{min} + rand(\boldsymbol{x}_{max} - \boldsymbol{x}_{min}) \tag{9.17}$$

$$\boldsymbol{v}_0^i = \frac{\boldsymbol{x}_{min} + rand(\boldsymbol{x}_{max} - \boldsymbol{x}_{min})}{\Delta t} = \frac{position}{time} \tag{9.18}$$

where $\boldsymbol{x_{min}}$ and $\boldsymbol{x_{max}}$ are the lower and upper bounds respectively.
At each iteration, the velocities and positions are updated as in the following equations (9.19)(9.20).

$$\boldsymbol{v}_{k+1}^i = w\boldsymbol{v}_k^i + c_1 rand\frac{p^i - \boldsymbol{x}_k^i}{\Delta t} + c_2 rand\frac{p_k^g - \boldsymbol{x}_k^i}{\Delta t} \tag{9.19}$$

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{x}_k^i + \boldsymbol{v}_{k+1}^i \Delta t \tag{9.20}$$

where, $\boldsymbol{v}_{\boldsymbol{k}+1}^i$ is the velocity of the particle i at time $\boldsymbol{k}+1$, $\boldsymbol{w}$ is the inertia factor, $\boldsymbol{wv_k^i}$ is the current motion, $\boldsymbol{c_1}$ is the self confidence, $\boldsymbol{c_1 rand\frac{p^i - x_k^i}{\Delta t}}$ is the particle memory influence, c_2 is the swarm confidence and $\boldsymbol{c_2 rand\frac{p_k^g - x_k^i}{\Delta t}}$ is the swarm influence. $\boldsymbol{p}^i$ is the pbest position of the particle i. Each particle keeps swimming in the coordinates solution space, which is associated with the best solution (fitness) that has been achieved. This value is called the personal best(pbest). Another $\boldsymbol{p^g}$ is the gbest position. The best value that is taken by the PSO, is the best value obtained by all particles. This value is called the global best (gbest).

9.3 Methodology

The above-introduced control algorithms were implemented using the SIMULINK environment in MATLAB. The quad-copter dynamics are shown in Table 9.1.
Quad-copter is a system which has six controllers for each degree of freedom, namely, pitch (φ), roll (θ) and yaw (ψ) Euler angles and X, Y and Z position coordinates. Hence, the initial system is designed to use six PID controllers $(PID_\varphi, PID_\theta, PID_\psi, PI_X, PI_Y, PID_Z)$.
The overall view of the PID-based quad-copter system is presented in Fig. 9.4. Note

Table 9.1: Model parameters of the Quad copter

Description	**Value**
Gravitational Acceleration (g)	$9.81ms^{-2}$
Quad copter mass (m)	$0.65\ kg$
Distance from center to motor (l)	$0.23m$
About x axis moment of inertia (jx)	$0.0075kgm^2$
About y axis moment of inertia (jy)	$0.0075kgm^2$
About z axis moment of inertia (jz)	$0.013kgm^2$
Force constant of propellers (k)	$0.0000313Ns^2$
Torque constant of propellers (d)	$0.00000075Ns^2$

Table 9.2: All PID parameters (Zeigler-Nichols)

	X	Y	Z	φ	θ	ψ
K_p	0.2	0.2	0.6	0.51	0.51	0.105
K_i	0.1	0.1	0.35	0.81	0.81	0.005
K_d	0.0	0.0	0.43	0.81	0.81	0.005

that, X and Y PI controllers are placed in cascaded with θ and ψ PID controllers, respectively.

The PID tuning was done using the Zeigler-Nichols method [28]. Each PID controller was tuned, starting with the altitude controller (Z) separately. Table 9.2 shows all tuned parameters used in the PID controllers.

Using the above-tuned PID parameters, the output response was collected against the input and recorded for the ANFIS algorithm training.

The recorded data were divided into two segments training and testing at the ratio of 70% and 30%. $Genfis2$ function in MATLAB is used as the initial membership function generation in FIS. The training was conducted offline, and then the error was observed during the testing phase.

GA and PSO were used to optimize the FISs which were initially generated using the ANFIS algorithm. These two optimization algorithms were used in the optimal parameter settings as mentioned in the literature [39]. The parameter setting of the GA and PSO is shown the Table 9.3 and 9.4.

Designing the GA-ANFIS and PSO-AFNIS controllers were performed as in the following pseudo-codes. Algorithm 4 shows the main steps in tuning the FIS parameters using GA. As introduced before, the three main operations were implemented to search for the best solution in the solution space. The overall representation of the GA-ANFIS algorithm in the quad-copter system is shown in Fig. 9.5.

Table 9.3: GA parameters

Iterations	100
Population Size	40
Crossover Percentage	50% Uniform crossover
Mutation Percentage	0.5%
Number of Mutants	20
Gamma (γ)	0.2
Mutation Rate (μ)	0.1
Selection Pressure (β)	8

Table 9.4: PSO parameters

Iterations	100
Population Size	40
Inertia Weight (w)	0.5
Inertia Weight Damping ratio	0.99
Personal Learning Coeffition (c_1)	1.5
Global Learning Coeffition (c_2)	1.5

Algorithm 4: GA-ANFIS optimization

```
Result: Optimized GA-FIS
Initial FIS;
begin
    Data acquisition from PID controllers;
    Random Partitioning of data as training and testing;
    Initial Fuzzy Inference System (FIS) generation using the training set;
end
GA tuning;
begin
    Selection;
    Crossover;
    Mutation;
end
Store the best cost and best solution;
Update the FIS;
```

The construction of PSO-ANFIS is presented in Algorithm 5. First, the initial FIS is generated using the ANFIS algorithm and then PSO is used to tune the FIS parameters furthermore. The overall representation of the PSO-ANFIS algorithm in the quad-copter system is shown in Fig. 9.6.

Algorithm 5: PSO-ANFIS optimization

Result: Optimized PSO-FIS

Initial FIS;

begin

  Data acquisition from PID controllers;

  Random Partitioning of data as training and testing;

  Initial Fuzzy Inference System (FIS) generation using the training set;

end

PSO tuning;

begin

  Generate Random Population;

  Velocity update;

  Position update;

end

Store the best cost and best solution;

Update the FIS;

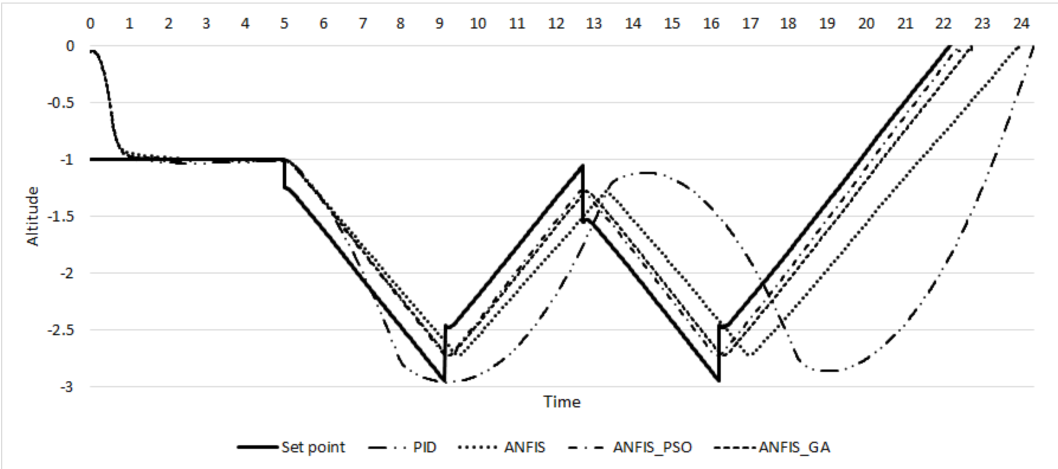


Figure 9.7: Altitude Response

Table 9.5: RMSE of each ANFIS controller

Error	Algorithm	Pitch	Roll$\times 10^{-5}$	Yaw$\times 10^{-8}$	Z
RMSE	**ANFIS**	0.000124	2.889	1.755	0.00434
	GA-ANFIS	0.000122	2.784	1.594	0.00337
	PSO-ANFIS	0.000112	2.782	1.585	0.00198

9.4 Results and Discussion

The experiments were conducted using two different simulations, altitude response measurement and way-point navigator, to determine the effectiveness of the corresponding control algorithms. The results were obtained for PID, ANFIS, GA-ANFIS and PSO-ANFIS.

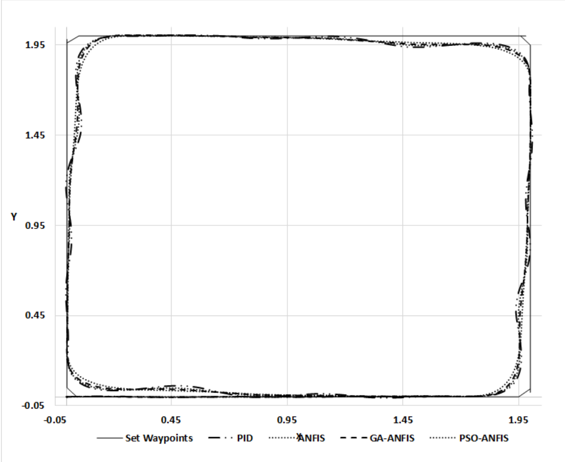


Figure 9.8: Way-point Follower

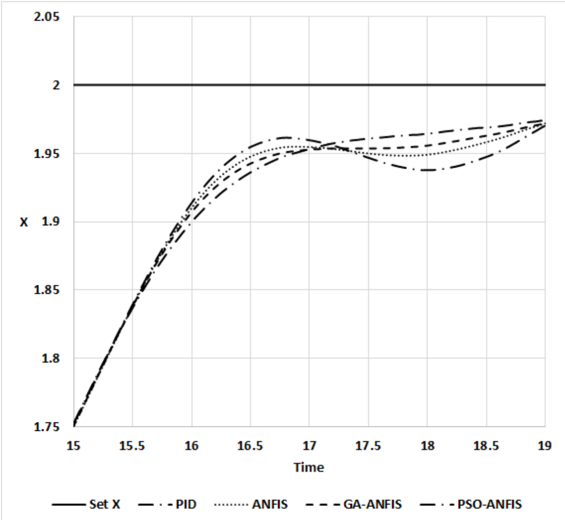


Figure 9.9: Change in X coordinates (I)

Table 9.6: Training time of each ANFIS controller

	Algorithm	Pitch	Roll	Yaw	Z
Time(s)	GA-ANFIS	705.32	696.32	705.90	697.97
	PSO-ANFIS	588.03	593.22	596.57	572.88

As the first simulation test, the altitude response was measured. A MATLAB simulation was designed to change the altitude for 22 seconds. The set coordinates were given in the z-axis, which is the altitude in this system. The results can be presented in Fig. 9.7. As shown in this figure, PID, ANFIS, PSO-ANFIS and GA-ANFIS performance in obtaining the desired setpoints are shown. When comparing the PID response with the other three ANFIS algorithms, the ANFIS-based systems show a smoother and more accurate response. It is also clearly mentioned that the response delay is much more

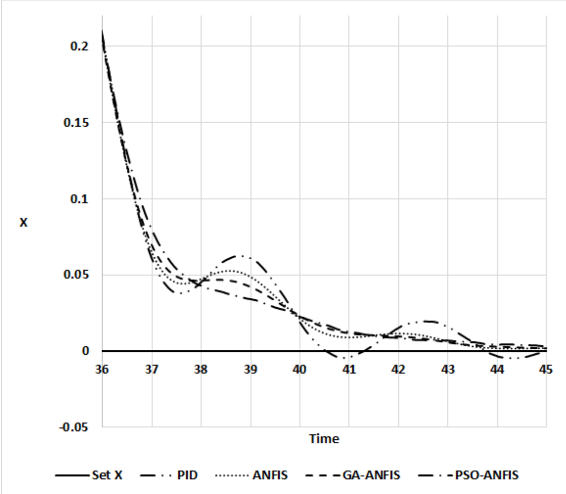


Figure 9.10: Change in X coordinates (II)

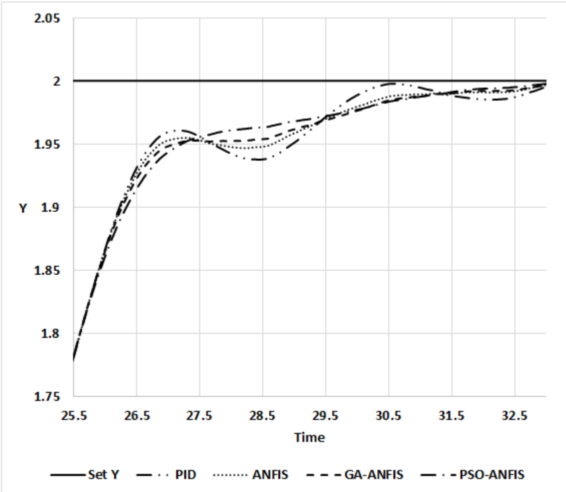


Figure 9.11: Change in Y coordinates (I)

significant in the PID controller than in ANFIS controllers.

Among the ANFIS controllers, such as ANFIS, GA-ANFIS and PSO-ANFIS, the best results were given by the PSO-ANFIS controller.

The next simulation was conducted to obtain the performance in X and Y axis navigation. In this simulation, five set points were defined as $(0, 0, -1), (2, 0, -1), (2, 2, -1), (0, 2, -1), (0, 0, -1)$ Here, the first, second and third coordinates are X, Y and Z, respectively. The altitude was kept constant at $1m$, and X and Y coordinates were changed accordingly. The results are presented using Figs. 9.8, 9.9, 9.10, 9.11, and 9.12.

Fig. 9.8 shows the total way-point navigation of the drone in X and Y planes. Fig. 9.9 and Fig. 9.10 are the closeup views of two instances in X coordinate variations. As in these figures, it is clear that the PSO-ANFIS performs better with less overshoot and faster stability. Y coordinate variations are shown in the Fig. 9.11 and Fig. 9.12. Here,

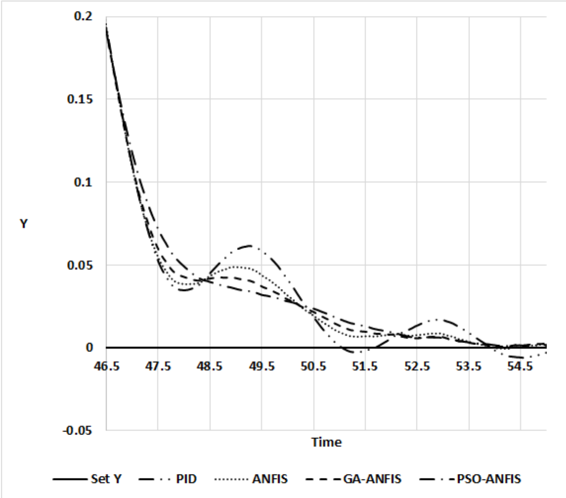


Figure 9.12: Change in Y coordinates (II)

the results are similar to the X coordinate variations. The numerical data for steady-state errors for the PID, ANFIS, GA-ANFIS and PSO-ANFIS are 0.00861, 0.00212, 0.00106, and 0.00009, respectively. The Rising times are $9s$, $8.73s$, $8.69s$ and $9.12s$ accordingly.

The results in Figs. 9.7, 9.8, 9.9, 9.10, 9.11, and 9.12 proves the superiority of PSO-ANFIS and GA-ANFIS over Conventional ANFIS and PID controllers. The nature-inspired algorithms (GA-ANFIS and PSO-ANFIS) perform better than the ANFIS and traditional PID controllers.

Moreover, Table 9.5 shows the Root Mean Squared Error (RMSE) for each controller. It can be observed that PSO-ANFIS and GA-ANFIS show very similar results. However, PSO has less error and faster stability than GA. This may be caused due to the following differences in the considered algorithms [40].

The inbuilt guidance mechanism in PSO causes faster convergence. But in GA, there is no dedicated guidance mechanism. Only when the particles participate in the crossover the better solutions converge. PSO handles two populations simultaneously, namely, *pbest* (personal best) and *current positions*. This key feature helps the PSO re-navigate in the solution space if the solution moves towards an unwanted path. GA does not handle two populations. Hence this becomes a massive advantage in solution exploration. The algorithm construction of the PSO is purely mathematical, while the GA has non-mathematical operations such as crossover operation. This gives the benefit of less time consumption in PSO than GA, and it can be seen in Table 9.6.

9.5 Conclusion and future works

In this paper, performance analysis is conducted to compare the results of the evolution of ANFIS controllers with optimization algorithms such as GA and PSO. A quad-copter is used here for the control application. We have also used the traditional PID controller to initially tune all other algorithms and then individually optimize the rest. The experiments were conducted as simulations. Two simulations as, altitude test and way-point follower, were used to obtain the response in X, Y and Z axis navigation. As shown in the results section in this paper, the PSO-ANFIS algorithm outperforms all the other control algorithms. Both simulations show that the PSO-ANFIS is better in quad-copter applications because it has lesser overshoot and stabilizes faster than the other algorithms used here. As for the future objective, the implementation in real-time and performance analysis can be indicated.

References

[1] Namal Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. Performance comparison of the anfis based quad-copter controller algorithms. In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2021.

[2] UM Rao Mogili and BBVL Deepak. Review on application of drone systems in precision agriculture. *Procedia computer science*, 133:502–509, 2018.

[3] G DEVI, N Sowmiya, K Yasoda, K Muthulakshmi, and K BALASUBRAMANIAN. Review on application of drones for crop health monitoring and spraying pesticides and fertilizer. *J. Crit. Rev*, 7:667–672, 2020.

[4] Nehemia Cohen. Aerial transport system, June 11 2009. US Patent App. 12/300,126.

[5] Fumiatsu Yakushiji, Koki Yakushiji, Mikio Murata, Naoki Hiroi, Keiji Takeda, and Hiroshi Fujita. The quality of blood is not affected by drone transport: An evidential study of the unmanned aerial vehicle conveyance of transfusion material in japan. *Drones*, 4(1):4, 2020.

[6] Balmukund Mishra, Deepak Garg, Pratik Narang, and Vipul Mishra. Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, 156:1–10, 2020.

[7] Rameesha Tariq, Maham Rahim, Nimra Aslam, Narmeen Bawany, and Ummay Faseeha. Dronaid: A smart human detection drone for rescue. In *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT)*, pages 33–37. IEEE, 2018.

[8] Henrike Schmidt. From a bird's eye perspective: Aerial drone photography and political protest. a case study of the bulgarian# resign movement 2013. *Digital Icons: Studies in Russian, Eurasian and Central European New Media*, 13:1–27, 2015.

[9] AK Puttock, AM Cunliffe, K Anderson, and Richard E Brazier. Aerial photography collected with a multirotor drone reveals impact of eurasian beaver reintroduction on ecosystem structure. *Journal of Unmanned Vehicle Systems*, 3(3):123–130, 2015.

[10] Jonathan Omega Escobedo-Alva, Elba Cinthya Garcia-Estrada, Luis Alberto Paramo-Carranza, Jesus Alberto Meda-Campana, and Ricardo Tapia-Herrera. Theoretical application of a hybrid observer on altitude tracking of quadrotor losing gps signal. *IEEE Access*, 6:76900–76908, 2018.

[11] Carlos Aguilar-Ibanez and Miguel S Suarez-Castanon. A trajectory planning based controller to regulate an uncertain 3d overhead crane system. *International Journal of Applied Mathematics and Computer Science*, 29(4), 2019.

[12] José Rafael García-Sánchez, Salvador Tavera-Mosqueda, Ramón Silva-Ortigoza, Victor Manuel Hernández-Guzmán, Jacobo Sandoval-Gutiérrez, Mariana Marcelino-Aranda, Hind Taud, and Magdalena Marciano-Melchor. Robust switched tracking control for wheeled mobile robots considering the actuators and drivers. *Sensors*, 18(12):4316, 2018.

[13] Dany Ivan Martinez, José Jesús De Rubio, Tomas Miguel Vargas, Victor Garcia, Genaro Ochoa, Ricardo Balcazar, David Ricardo Cruz, Arturo Aguilar, Juan Francisco Novoa, and Carlos Aguilar-Ibanez. Stabilization of robots with a regulator containing the sigmoid mapping. *IEEE Access*, 8:89479–89488, 2020.

[14] Dany Ivan Martinez, José de Jesús Rubio, Victor Garcia, Tomas Miguel Vargas, Marco Antonio Islas, Jaime Pacheco, Guadalupe Juliana Gutierrez, Jesus Alberto Meda-Campaña, Dante Mujica-Vargas, and Carlos Aguilar-Ibañez. Transformed

structural properties method to determine the controllability and observability of robots. *Applied Sciences*, 11(7):3082, 2021.

[15] Dany Ivan Martinez, José de Jesús Rubio, Arturo Aguilar, Jaime Pacheco, Guadalupe Juliana Gutierrez, Victor Garcia, Tomas Miguel Vargas, Genaro Ochoa, David Ricardo Cruz, and Cesar Felipe Juarez. Stabilization of two electricity generators. *Complexity*, 2020, 2020.

[16] M Heryanto, Herwin Suprijono, Bhakti Yudho Suprapto, and Benyamin Kusumoputro. Attitude and altitude control of a quadcopter using neural network based direct inverse control scheme. *Advanced Science Letters*, 23(5):4060–4064, 2017.

[17] Gil-Young Yoon, Akito Yamamoto, and Hun-Ok Lim. Mechanism and neural network based on pid control of quadcopter. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pages 19–24. IEEE, 2016.

[18] Zaid Tahir, Mohsin Jamil, Saad Ali Liaqat, Lubva Mubarak, Waleed Tahir, and Syed Omer Gilani. Design and development of optimal control system for quad copter uav. *Indian Journal of Science and Technology*, 9(25):10–17485, 2016.

[19] Halima Housny, Hassan El Fadil, et al. Pso-based anfis for quadrotor system trajectory-tracking control. In *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pages 1–6. IEEE, 2020.

[20] Boumediene Selma, Samira Chouraqui, and Hassane Abouaïssa. Hybrid anfis-ant colony based optimisation for quadrotor trajectory tracking control. *International Journal of Modelling, Identification and Control*, 34(1):13–25, 2020.

[21] Mohammad Al-Fetyani, Mohammad Hayajneh, and Adham Alsharkawi. Design of an executable anfis-based control system to improve the attitude and altitude performances of a quadcopter drone. *International Journal of Automation and Computing*, pages 1–17, 2020.

[22] Md Belal Hossen and Bashudeb Chandra Ghosh. Performance analysis of a pmbldc motor drive based on anfis controller and pi controller. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6. IEEE, 2019.

[23] Shahida Khatoon, Ibraheem Nasiruddin, and Mohammad Shahid. Design and simulation of a hybrid pd-anfis controller for attitude tracking control of a quadrotor uav. *Arabian Journal for Science and Engineering*, 42(12):5211–5229, 2017.

[24] Boumediene Selma, Samira Chouraqui, and Hassane Abouaïssa. Optimal trajectory tracking control of unmanned aerial vehicle using anfis-ipso system. *International Journal of Information Technology*, 12(2):383–395, 2020.

[25] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics and Automation magazine*, 19(3):20–32, 2012.

[26] Terrence L Blevins. Pid advances in industrial control. *IFAC Proceedings Volumes*, 45(3):23–28, 2012.

[27] Narong Aphiratsakun and Nane Otaryan. Ball on the plate model based on pid tuning methods. In *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–4. IEEE, 2016.

[28] PM Meshram and Rohit G Kanojiya. Tuning of pid controller using ziegler-nichols method for speed control of dc motor. In *IEEE-international conference on advances in engineering, science and management (ICAESM-2012)*, pages 117–122. IEEE, 2012.

[29] Ahmad Aftas Azman, Mohd Hezri Fazalul Rahiman, Nurul Nadia Mohammad, Mohd Hezri Marzaki, Mohd Nasir Taib, and Mohd Fozi Ali. Modeling and comparative study of pid ziegler nichols (zn) and cohen-coon (cc) tuning method for multi-tube aluminum sulphate water filter (mtas). In *2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pages 25–30. IEEE, 2017.

[30] S Anusha, G Karpagam, and E Bhuvaneswarri. Comparison of tuning methods of pid controller. *International Journal of Management, Information Technology and Engineering*, 2(8):1–8, 2014.

[31] Tepljakov Aleksei, Petlenkov Eduard, and Belikov Juri. A flexible matlab tool for optimal fractional-order pid controller design subject to specifications. In *Proceedings of the 31st Chinese Control Conference*, pages 4698–4703. IEEE, 2012.

[32] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[33] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.

[34] Mehmet Şahin and Rızvan Erol. A comparative study of neural networks and anfis for forecasting attendance rate of soccer games. *Mathematical and Computational Applications*, 22(4):43, 2017.

[35] Jasbir Singh Arora. *Introduction to optimum design*. Elsevier, 2004.

[36] Huu Khoa Tran and Thanh Nam Nguyen. Flight motion controller design using genetic algorithm for a quadcopter. *Measurement and Control*, 51(3-4):59–64, 2018.

[37] Seyedali Mirjalili. Genetic algorithm. In *Evolutionary algorithms and neural networks*, pages 43–55. Springer, 2019.

[38] James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, volume 5, pages 4104–4108. IEEE, 1997.

[39] Rania Hassan, Babak Cohanim, Olivier De Weck, and Gerhard Venter. A comparison of particle swarm optimization and the genetic algorithm. In *46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, page 1897, 2005.

[40] Chao Ou and Weixing Lin. Comparison between pso and ga for parameters optimization of pid controller. In *2006 International conference on mechatronics and automation*, pages 2471–2475. IEEE, 2006.

Chapter 10

Effective attempt to recognize hand gestures using gradient boosting algorithms.

This work is presented at International Symposium on Information and Communication Technology (ISoICT 2022) [1].

10.1 Introduction

Humans' need for and degree of demand for services grows every day. Hand gestures and human body language both contribute significantly to face-to-face communication. Most explanations in communication use hand gestures, and studying them can provide us with some understanding of how we communicate. Hand gestures are crucial when communication between a hearing person and a deaf person is established [2]. However, as mentioned in [3, 4], existing automation in this field does not prioritize using hand gestures in routine operations. These hand gestures can help operate household appliances [5]. We are headed for a day when hand gestures will be able to handle everything. The user is now given the complexity of the numerous computer programs and user interfaces' functions due to the rapid advancement of technology in this field. Image processing is currently employed to simplify and make this system easier to grasp. It is necessary to enter hand gesture images into the system and conduct further analysis to determine their meaning [6].

Several cutting-edge algorithms, like Histogram of Oriented Gradients (HOG), CNN, and Bagging, may be used to ensure good outcomes. For classification, algorithms like k-Nearest Neighbours (KNN), logistic regression, Support Vector Classifier (SVC), Naive Bayes, and stochastic gradient descent are also available. Recent works on machine learning are available in several gradient boosting methods, such as XGBoost, CatBoost, and LightGBM, which perform far better than the black box algorithms in some applications. Compared with the CNN methods, gradient boosting methods shows faster response and calculation speed.

Fuzzy-based algorithms are considered to provide results that can imitate human reasoning. Fuzzy logic shows remarkable performances in dealing with nonlinear, uncertain variables. Therefore, fuzzy has been used in many real-time applications in recent years to improve system performances. Therefore, this study aims to develop a novel ensemble stacking method combining gradient boosting methods and the ANFIS algorithm.

10.2 Related Works

Numerous types of research have been conducted on hand gesture recognition in the past. Heo et al. proposed the Binary open and close technique on a hand gesture dataset and employed the system for game implementation [7]. In 2011, Dardas and Georganas researched a real-time dataset of hand gesture classification. They achieved 96.23% accuracy using Support Vector Machines (SVM), K-means clustering, and Scale Invariant Feature Transform (SIFT) features [8]. Zheng et al. proposed a method using a Three-axis accelerometer (ACC) for classification and obtained 95.3% accuracy [9].

The classification of hand gesture studies varies depending on the sensors used in the research experiment. Most of the researchers tend to use images, and others try to use motion sensors. Kinect is an image sensor with a depth camera. Ren et al. and Plouffe and Cretu employed Kinect sensors for image acquisition of hand gestures and obtained 93.2% and 92.4% accuracies, respectively [10–12].

CNN methods are well-known for classification tasks. Several approaches used CNN methods in hand gesture classification. Zhang et al. employed CNN and Long Short-Term Memory (LSTM) to develop a classification system and obtained 95.72% for the Jester dataset and 95.69% accuracy for the Nvidia dataset [13]. Zhao and Wang conducted a classification experiment on American Sign Language (ASL) dataset using the

CNN algorithm [14]. Tam et al. achieved 98.2% accuracy for the Nina dataset and real-time hand gesture dataset using CNN and a myoelectric control scheme [15].
Fuzzy logic-based methods are widely used in human-like classification tasks. There are several attempts at the classification of hand gestures using fuzzy logic. Li et al. proposed Spatial Fuzzy Matching (SFM) with leap motion for classification tasks and achieved 94% to 100% accuracy for static gestures and 90% for dynamic gestures [16]. Hira et al. employed fuzzy logic as an optimization algorithm in the classification research and obtained 88.46% and 87.69% accuracies for IPN hand and Jester datasets, respectively [17]. A classification method is proposed by Tong et al. using Fuzzy Gaussian mixture models (FGMMs), and the experiment results conclude that the implemented system is effective in real-time systems.

10.3 Methodology

10.3.1 Gradient Boosting Methods

One kind of machine learning boosting is gradient boosting. It is predicated on the hunch that the overall prediction error is minimized when prior models are coupled with the best feasible upcoming model. Setting the desired results for this subsequent model to reduce mistakes is essential. Each case's goal result will vary depending on how much altering a case's forecast affects the total prediction error.
Because target outcomes are determined for each case based on the gradient of the error concerning the prediction, this method is known as gradient boosting. In the space of potential predictions for each training instance, each new model moves in a direction that minimizes prediction error [18].

10.3.1.1 XGBoost

The XGBoost algorithm was created by Chen et al. [19]. It is an innovative and expandable use of gradient-boosting machines that have been shown to push the computational capacity of boosted tree algorithms. It was created exclusively to improve the model performance and computational efficiency.
Adding additional models as part of an ensemble approach known as "boosting" corrects faults created by earlier models. In the method known as gradient boosting, new models have been generated that forecast the residuals of older models, which are then combined

to get the final prediction. Recursively adding models continues until no discernible progress is made. When incorporating new models, the loss is minimized using a gradient descent approach.

By 2015, XGBoost had completed 17 of the 29 Machine Learning (ML) projects that had been put on Kaggle. Using many CPU cores and cutting the look-up times of individual trees made with XGBoost considerably increased speed. This approach uses novel regularisation techniques and is built in R and Python using the SciKit-Learn [20] package.

10.3.1.2 CatBoost

A potent machine learning approach called gradient boosting can handle challenges with diverse features, noisy data, and complicated relationships. Yandex developers presented CatBoost, a machine learning technique based on gradient boosting decision trees (GBDT), in 2017. [21]. The following benefits of CatBoost over other GBDT algorithms:

1. The algorithm does a good job of handling categorical features. Categorical characteristics can be replaced with appropriate average label values using conventional GBDT techniques.

2. Multiple category characteristics are combined in CatBoost. All categorical characteristics and combinations in the current tree are combined with all categorical features in the dataset using a greedy method by CatBoost.

3. Gradient bias may be addressed with CatBoost. In GBDT, a weak learner is generated with each iteration, and each learner is trained using the gradient of the previous learner. The output is the sum of all learners' categorized results [22].

10.3.1.3 LightGBM

Microsoft's LightGBM [23] is an open-source GBDT algorithm. The parallel voting decision tree technique, which accelerates training, consumes less memory, and combines sophisticated network connectivity to maximize parallel learning, is based on the histogram-based algorithm [24, 25].

Divide the training data across several machines. Each iteration, decide on the local voting decision to choose the top k characteristics and the global voting decision to

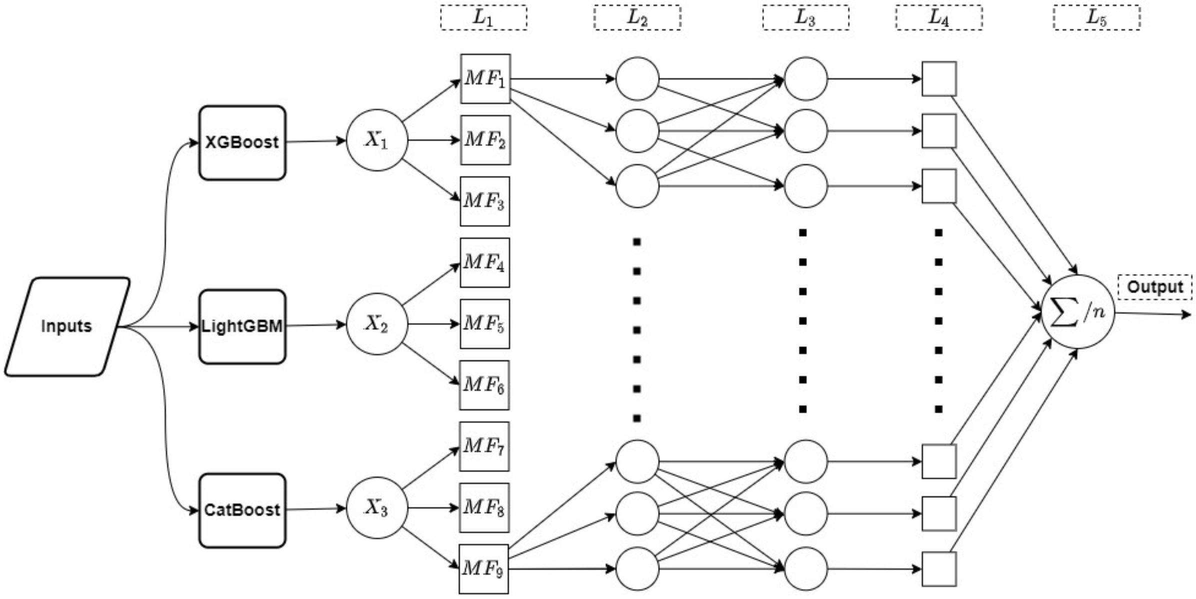


Figure 10.1: Proposed Ensemble model structure with ANFIS and Gradient Boosting algorithms.

receive the top 2k attributes. To identify the leaf with the most significant splitter gain, LightGBM employs the leaf-wise approach.

10.3.2 Ensemble Model Design

The Designing of the ensemble structure is shown in Figure 10.1. As mentioned above, the dataset consists of 26 input data points used for the initial phase of gradient boosting training. At the initial phase, XGBoost, CatBoost, and LightGBM are used as individual components and trained in parallel. Each of the gradient-boosting algorithms generates outputs corresponding to the input dataset.

The number of outputs at the second stage of the ensemble model is three. Therefore, the ANFIS model is designed with three inputs and one output, as shown in Figure 10.1. Let the inputs be X_1, X_2, and X_3. Three memberships were generated using bell membership functions. The calculation of bell memberships for each input can be presented by Equation 10.1.

$$\mu_{Ai}(x) = \frac{1}{1 + \left\{\left(\frac{x-c_i}{a_i}\right)^2\right\}^{b_i}} \tag{10.1}$$

Here, membership function parameters are a_i, b_i, and c_i.

Table 10.1: Machine learning algorithm parameters.

Algorithm	Parameter	Value
XGBOOST	objective	reg:linear
	colsample_bytree	0.3
	learning_rate	0.01
	max_depth	5
	Alpha	10
	n_estimators	10
CatBoost	Iterations	500
	learning_rate	0.01
	eval_metric	MultiClass
	sampling_frequency	PerTree
	penalties_coefficient	1
	max_leaves	64
	permutation_count	4
	Depth	6
LightGBM	num_leaves	31
	objective	binary
	learning_rate	0.01
	boosting_type	Dart
ANFIS	Membership_Function	Bell
	Number_of_MFs	3
	Number_of_Inputs	3
	Iterations	100

As in Figure 10.1, $MF_1, MF_2, ..., MF_9$ are the memberships generated for the three inputs. As described in the introduction of ANFIS, the general ANFIS calculations are performed along the proceeding layers.

The parameters used in this study for each algorithm can be presented using Table 10.1. All the parameters were tuned by repeating the experiments multiple times until the best accuracy was gained.

10.4 Results and Discussion

10.4.1 Dataset

The hand gesture dataset used in this experiment is explained as follows. Five participants performed seven distinct motions every ten times, yielding 50 samples for each gesture across three separate datasets [26]. However, this study considered the dataset in which the frame sequences are combined to generate a single feature file for each sample. The number of frames used in this case is 30, and motion-based normalization has been considered to generate the data file. This dataset is a combination of 350 samples and seven classes: Move down, Move left, Move right, Move up, Press, Zoom in, and Zoom out.

A gesture is made up of a series of related frames. Each gesture mentioned above is presented using either one or both hands. Zoom-in and Zoom-out classes were represented using both hands, while other classes used single-hand movements.

The features are based on the centroid of the hand palm, the fingertip locations, and the depth value. The inputs of the dataset can be presented as follows [26].

- Left centroid position (LC)
- Left centroid depth (LCD)
- Right fingers positions numbered clockwise (RF1, RF2, RF3, RF4, RF5)
- Frame number (FN)
- Left fingers positions numbered clockwise (LF1, LF2, LF3, LF4, LF5)
- Right centroid position (RC)
- Right centroid depth (RCD)
- Gesture number (GN)

However, there are numerous hand gesture datasets available in the scientific world. Most of these datasets are generated purely based on the image [27–31], and some of the datasets use electronic positioning techniques such as accelerometer, gyroscope, and compass to extract the feature set [32–34]. Image datasets naturally require preprocessing and substantial computational power for model training. When compared with the image and numerical data, it is known that the numerical datasets can be trained and utilized faster than image-based datasets. Moreover, there can be limitations due to lightning conditions when images are used as the data for hand gesture recognition. Hence, above mentioned numerical dataset that was constructed using the locations of hands and palms is used in this study to construct the classification models. The current study employed a 7:3 ratio of data samples for training the model and testing, respectively.

The experiment results were analyzed using two different methods such as confusion matrix-based and K-Fold cross-validation.

Table 10.2: Classification Report for XGBoost, CatBoost, LightGBM, and Proposed ensemble algorithms.

	Precision				Recall				F1-Score			
	XGBoost	CatBoost	LightGBM	Proposed	XGBoost	CatBoost	LightGBM	Proposed	XGBoost	CatBoost	LightGBM	Proposed
MoveDown	1.00	1.00	1.00	1.00	1.00	0.83	1.00	1.00	1.00	0.91	1.00	1.00
MoveLeft	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.89	1.00	1.00
MoveRight	1.00	1.00	0.36	1.00	1.00	1.00	1.00	0.80	1.00	1.00	0.53	0.89
MoveUp	1.00	0.85	1.00	0.92	0.78	0.92	0.58	1.00	0.88	0.88	0.74	0.96
Press	1.00	0.93	1.00	1.00	1.00	1.00	0.62	1.00	1.00	0.96	0.76	1.00
ZoomIn	0.57	0.89	1.00	1.00	1.00	1.00	1.00	1.00	0.73	0.94	1.00	1.00
ZoomOut	1.00	1.00	0.85	1.00	0.67	0.75	0.92	1.00	0.80	0.86	0.88	1.00
Accuracy									0.91	0.91	0.84	0.99
MicroAVG	0.94	0.92	0.89	0.99	0.92	0.93	0.87	0.97	0.91	0.92	0.84	0.98
WeightedAVG	0.95	0.92	0.93	0.99	0.91	0.91	0.84	0.99	0.92	0.91	0.86	0.99

10.4.2 K-Fold Cross Validation analysis

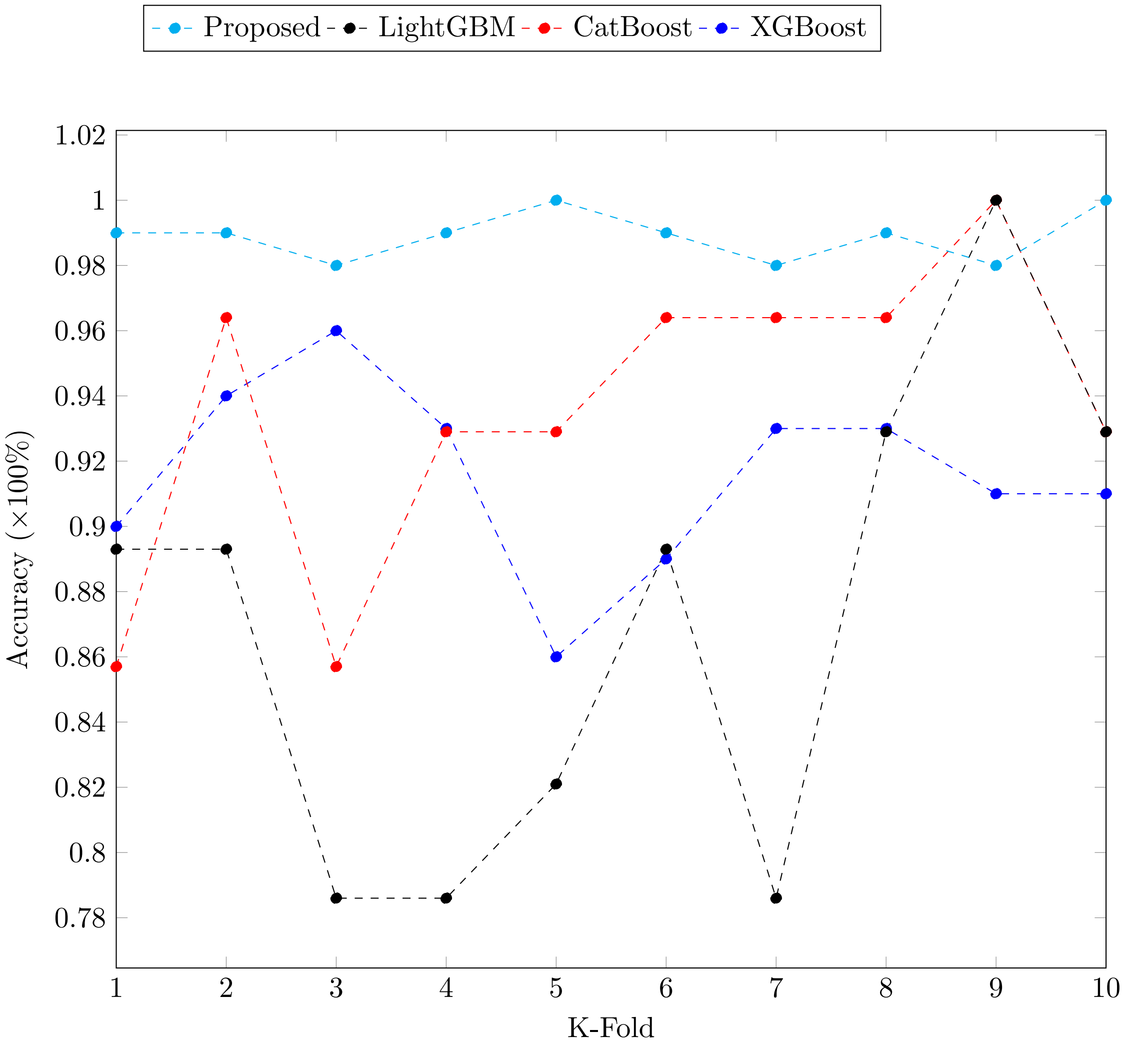


Figure 10.2: 10-Fold Cross-Validation of the Accuracy of the Classifications.

The K-Fold cross-validation was done on each algorithm and checked the reliability. As shown in Figure 10.2, the proposed algorithm showed the best accuracies closer to 100% at all folds. During folds 5 and 10, the testing accuracy reached 100%, while

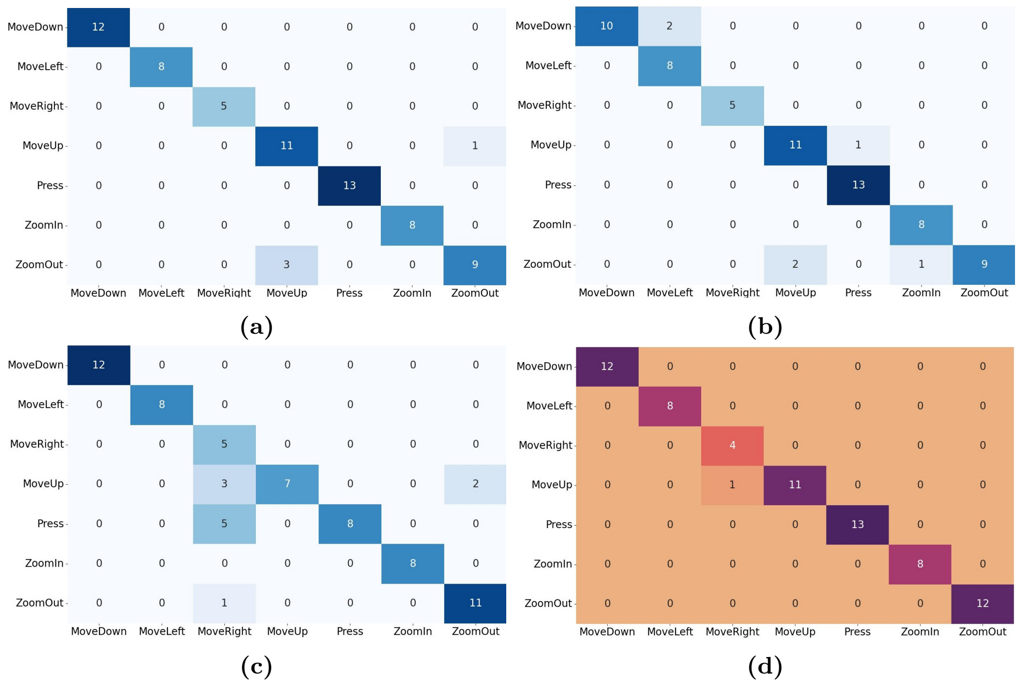


Figure 10.3: Confusion Matrix-based presentation of results: (a) XGBoost; (b) CatBoost; (c) LightGBM; (d) Proposed Algorithm.

in other instances, the accuracy remained more than 95%. Compared with the other algorithms, LightGBM shows varying accuracy during the 10-fold. The best accuracy of the LightGBM has reached the ninth fold, and it is 100%. However, in some instances, the accuracy drops to less than 80%. CatBoost and XGBoost algorithms performed better than LightGBM, as presented in the figure. The accuracies of the algorithms were kept at more than 85%. Moreover, according to the results, the CatBoost algorithm shows better results in 10-fold cross-validation than the XGBoost or LightGBM. Overall, the proposed algorithm outperformed the other three algorithms in accuracy comparison. The average accuracy for XGBoost, CatBoost, LightGBM, and the proposed algorithm were 91.60%, 93.57%, 87.16%, and 98.9%, respectively.

10.4.3 Confusion Matrix-based analysis

As shown in Figure 10.3, the confusion matrix of each algorithm alone and the ensemble model can be presented. XGBoost, CatBoost, LightGBM, and proposed algorithm result matrixes are shown in Figure 10.3a, 10.3b, 10.3c, and 10.3d respectively. The results show that the proposed algorithm classification outperforms other algorithms used in this study. It can be seen that the proposed algorithm has done one miss classification at Move right with Move up while the other confusion matrixes show more miss classifications.

Table 10.2 shows the precision, recall, and f1-score calculations for each algorithm confusion matrixes. These parameters were calculated considering each class individually. The supporting samples were considered the same amount for each algorithm. Here, MicroAVG and WeightedAVG stand for micro average and weighted average.

The table shows that the best micro average precision was obtained by the proposed algorithm having 0.99. The other algorithms have decreasing precision having 0.94, 0.92, and 0.89 for XGBoost, CatBoost, and LightGBM, respectively. When comparing the recall performance, the proposed algorithm poses 0.97, while other algorithms have a lesser recall. However, it can be noticed that the recall of CatBoost is higher than XGBoost, having 0.93 and 0.92, respectively. The LightGBM scored the minor recall with 0.87. Here, the micro average recall was considered for the comparison.

The f1-score of the proposed algorithm is 0.98, while CatBoost, XGBoost, and LightGBM showed lower f1-score with 0.92, 0.91, and 0.84, respectively. However, when considering the weighted f1-score, CatBoost has a higher f1-score than XGBoost.

According to the table, the proposed algorithm's accuracy outperforms the XGBoost, CatBoost, and LightGBM by subsequently having 0.99, 0.91, 0.91, and 0.84. The table shows that the accuracy of XGBoost and CatBoost have the same value (0.91).

10.4.4 Conclusion

Most people consider Sign Language the universal language due to the similarities in the motion to express the idea. There have been numerous methods and applications developed to recognize hand gestures in the past. However, most of these methods use BlackBox algorithms, such as CNN-based methods. Although past studies showed high accuracies, implementing these models in low-resource platforms is problematic. Therefore, this study aimed to construct an efficient, accurate, and low-weight model for hand gesture recognition using frame-based datasets. The dataset used in this study is a combination of seven classes of hand gestures. Each dataset sample was constructed by summing 30 frames of the gesture motion.

XGBoost, CatBoost, and LightGBM are trending gradient boosting algorithms, and they have succeeded in showing outstanding performances in classification with low model weight. Moreover, these algorithms are high-speed compared to the CNN-based methods, do not require GPU, and can be trained using only CPU processing. Hence,

this study considered three gradient boosting methods along with the ANFIS algorithm to develop an ensemble boosting algorithm. ANFIS is a fuzzy logic-based algorithm that can imitate human thinking due to the membership function involvements. However, due to the computational complexity, ANFIS limits use more inputs to train a model. Therefore, the current study used gradient-boosting methods for the initial step of the ensemble model and then used the ANFIS to boost the accuracies.
The results of this study show promising performance for the proposed ensemble model with higher accuracies than XGBoost, CatBoost, and LigthGBM. The results were compared using 10-fold cross-validation to verify the performance. The 10-fold cross-validation guarantees that the proposed algorithm outperformed other algorithms having high accuracy and effectiveness. As for the future perspective, there are several pathways. The dataset used in the paper is small compared with other datasets for hand gesture recognition such as IPN[35], EgoGesture [28]. Therefore, the robustness of the proposed method can be further evaluated. Moreover, this method can be used to develop real-time hand gesture recognition and implemented in a hardware-level platform such as Field-programmable Gate Array (FPGA) for faster response.

References

[1] Namal Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. Designing and implementation of novel ensemble model based on anfis and gradient boosting methods for hand gestures classification. In *The 11th International Symposium on Information and Communication Technology*, pages 283–289, 2022.

[2] Debasrita Chakraborty, Deepankar Garg, Ashish Ghosh, and Jonathan H Chan. Trigger detection system for american sign language using deep convolutional neural networks. In *Proceedings of the 10th International Conference on Advances in Information Technology*, pages 1–6, 2018.

[3] Sruthy Skaria, Akram Al-Hourani, and Robin J Evans. Deep-learning methods for hand-gesture recognition using ultra-wideband radar. *IEEE Access*, 8:203580–203590, 2020.

[4] Nasser H Dardas and Nicolas D Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Transactions on Instrumentation and measurement*, 60(11):3592–3607, 2011.

[5] Cem Keskin, Furkan Kirac, Yunus Emre Kara, and Laie Akarun. Randomized decision forests for static and dynamic hand shape classification. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 31–36. IEEE, 2012.

[6] Sy Bor Wang, Ariadna Quattoni, L-P Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1521–1527. IEEE, 2006.

[7] Hwan Heo, Eui Chul Lee, Kang Ryoung Park, Chi Jung Kim, and Mincheol Whang. A realistic game system using multi-modal user interfaces. *IEEE Transactions on Consumer Electronics*, 56(3):1364–1372, 2010.

[8] Nasser Dardas, Qing Chen, Nicolas D Georganas, and Emil M Petriu. Hand gesture recognition using bag-of-features and multi-class support vector machine. In *2010 IEEE International Symposium on Haptic Audio Visual Environments and Games*, pages 1–5. IEEE, 2010.

[9] Xu Zhang, Xiang Chen, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang. A framework for hand gesture recognition based on accelerometer and emg sensors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1064–1076, 2011.

[10] Guillaume Plouffe and Ana-Maria Cretu. Static and dynamic hand gesture recognition in depth data using dynamic time warping. *IEEE transactions on instrumentation and measurement*, 65(2):305–316, 2015.

[11] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[12] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia*, 15 (5):1110–1120, 2013.

[13] Wenjin Zhang, Jiacun Wang, and Fangping Lan. Dynamic hand gesture recognition based on short-term sampling neural networks. *IEEE/CAA Journal of Automatica Sinica*, 8(1):110–120, 2020.

[14] Yiwen Zhao and Lidan Wang. The application of convolution neural networks in sign language recognition. In *2018 ninth international conference on intelligent control and information processing (ICICIP)*, pages 269–272. IEEE, 2018.

[15] Simon Tam, Mounir Boukadoum, Alexandre Campeau-Lecours, and Benoit Gosselin. A fully embedded adaptive real-time hand gesture classifier leveraging hd-semg and deep learning. *IEEE transactions on biomedical circuits and systems*, 14 (2):232–243, 2019.

[16] Hua Li, Lifan Wu, Huan Wang, Cheng Han, Wei Quan, and Jianping Zhao. Hand gesture recognition enhancement based on spatial fuzzy matching in leap motion. *IEEE Transactions on Industrial Informatics*, 16(3):1885–1894, 2019.

[17] Hira Ansar, Amel Ksibi, Ahmad Jalal, Mohammad Shorfuzzaman, Abdulmajeed Alsufyani, Suliman A Alsuhibany, and Jeongmin Park. Dynamic hand gesture recognition for smart lifecare routines via k-ary tree hashing classifier. *Applied Sciences*, 12(13):6481, 2022.

[18] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3):1937–1967, 2021.

[19] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[21] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[22] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[23] Qi Meng, Guolin Ke, Taifeng Wang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, and Tie-Yan Liu. A communication-efficient parallel algorithm for decision tree. *Advances in Neural Information Processing Systems*, 29, 2016.

[24] Sanjay Ranka and Vineet Singh. Clouds: A decision tree classifier for large datasets. In *Proceedings of the 4th knowledge discovery and data mining conference*, volume 2, 1998.

[25] Ruoming Jin and Gagan Agrawal. Communication and memory efficient parallel decision tree construction. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 119–129. SIAM, 2003.

[26] Sergiu oprea. Sergiuoprea/hand-gesture-recognition-datasets, Jan 2017. URL `https://github.com/sergiuoprea/Hand-Gesture-Recognition-Datasets`.

[27] Tuan Linh Dang, Huu Thang Nguyen, Duc Manh Dao, Hoang Vu Nguyen, Duc Long Luong, Ba Tuan Nguyen, Suntae Kim, and Nicolas Monet. Shape: a dataset for hand gesture recognition. *Neural Computing and Applications*, pages 1–14, 2022.

[28] Yifan Zhang, Congqi Cao, Jian Cheng, and Hanqing Lu. Egogesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia*, 20(5):1038–1050, 2018.

[29] Pramod Kumar Pisharady and Martin Saerbeck. Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, 141:152–165, 2015.

[30] Chong Wang, Zhong Liu, and Shing-Chow Chan. Superpixel-based hand gesture recognition with kinect depth camera. *IEEE transactions on multimedia*, 17(1): 29–39, 2014.

[31] Vinh Truong Hoang. Hgm-4: A new multi-cameras dataset for hand gesture recognition. *Data in brief*, 30:105676, 2020.

[32] Piotr Kaczmarek, Tomasz Mańkowski, and Jakub Tomczyński. putemg—a surface electromyography hand gesture recognition dataset. *Sensors*, 19(16):3548, 2019.

[33] Hari Prabhat Gupta, Haresh S Chudgar, Siddhartha Mukherjee, Tanima Dutta, and Kulwant Sharma. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, 16(16):6425–6432, 2016.

[34] Safa Ameur, Anouar Ben Khalifa, and Med Salim Bouhlel. A novel hybrid bidirectional unidirectional lstm network for dynamic hand gesture recognition with leap motion. *Entertainment Computing*, 35:100373, 2020.

[35] Gibran Benitez-Garcia, Jesus Olivares-Mercado, Gabriel Sanchez-Perez, and Keiji Yanai. Ipn hand: A video dataset and benchmark for real-time continuous hand gesture recognition. In *25th International Conference on Pattern Recognition, ICPR 2020, Milan, Italy, Jan 10–15, 2021*, pages 4340–4347. IEEE, 2021.

Appendix A

Supplementary information for Chapter 6

Table A.1: Descriptive Analysis of the Monobe River dataset

	Data Points	Mean	STD	25%	50%	75%	Max
Agriculture, forestry and fishery experience training center	3058	112.61	346.65	0	0	60	5180
Kahokucho Birafu	3058	117.62	370.42	0	0	57.5	5480
Kahokucho Nishigawa	3058	85.65	253.99	0	0	30	3590
Monobecho Yasumaru	3058	92.97	275.69	0	0	40	3520
Kahokucho Seizume	3058	91.5	267.03	0	0	40	3680
Monobecho Odochi	3058	92.41	272.67	0	0	40	3780
Kahokucho Birafu	3058	86.34	247.63	0	0	40	3140
Monobe	3058	64.92	179.5	0	0	25	1955
Water Level	3058	64.21	21.15	51	61	74	263

Table A.2: Descriptive Analysis of the Niyodo River dataset

	Data Points	Mean	STD	25%	50%	75%	Max
Kochi Prefecture Middle-West Civil Engineering Office	3139	82.01	245.05	0	0	30	4160
1931 Harunoch Saibata	3139	69.99	200.03	0	0	30	4130
2228 Ochik	3139	88.23	271.48	0	0	40	4440
77 Nakagumi	3139	88.98	272.8	0	0	40	5170
2036 Tokoroyama	3139	84.21	257.81	0	0	40	4800
617 Kaminanokawa	3139	97.3	289.03	0	0	60	4290
Kamikuroiwa	3139	61.07	177.8	0	0	40	2850
Kurofujigawa	3139	75.26	247.13	0	0	40	3920
4453-2 Hinoura	3139	76.86	217.12	0	0	60	3220
Nakatsu Hanamomo no Sato	3139	79.49	259.85	0	0	40	3970
3815 Takase	3139	83.72	281.43	0	0	40	5040
Osaki	3139	83.95	274.44	0	0	40	3950
Sakawa Municipal Kohoku Hospital	3139	87.27	264.51	0	0	40	4915
1076 hue	3139	76.55	300.01	0	0	10	5455
Water Level	3139	199.69	82.85	173	181	199	999

Table A.3: Descriptive Analysis of the Thu Bon River dataset

	Data Points	Mean	STD	25%	50%	75%	Max
Phuoc Son	778	12.57	32.30	0	0	9	320
Thanh My	778	9.05	50.08	0	0	2	1189
Hoi Khach	778	8.87	32.00	0	0	3	473
Ai Nghia	778	10.12	34.78	0	0	3	500
Tra My	778	16.95	45.62	0	0	9	408
Tien Phuoc	778	13.34	38.69	0	0	7	412
Hiep Duc	778	11.04	32.34	0	0	4.75	293
Nong Son	778	11.76	36.37	0	0	5	428
Hoi An	778	10.45	35.45	0	0	4	473
Water Level	778	66.28	52.95	36	52.5	78	552

Table A.4: Descriptive Analysis of the Kelani River dataset

	Data Points	Mean	STD	25%	50%	75%	Max
Kitulgala	5201	11.89	22.26	0	1.9	14.6	336.4
Holombuwa	5201	8.01	17.58	0	0.3	7.7	248.4
Deraniyagala	5201	11.55	22.33	0	1.6	14	355.6
Hanwella	5201	8.26	17.54	0	0	8.2	289.6
Colombo	5201	6.60	17.14	0	0	5.1045	440.2
Rathmalana	5201	6.90	17.16	0	0.2	5.8	382.8
Water Level	5201	1.34	0.52	1.06	1.24	1.47	7.44

Table A.5: Descriptive Analysis of the Kalu River dataset

	Data Points	Mean	STD	25%	50%	75%	Max
Alupolla	5843	12.02	20.09	0	2.7	16.2	186.2
Balangoda PO	5843	6.02	13.00	0	0	6	150
Galathura	5843	10.28	17.76	0	2	14	297
Halwathura	5843	11.55	20.42	0	0	16	257
Ratnapura	5843	10.04	18.33	0	1.6	12.8	345.2
Horana	5843	8.05	16.53	0	0	10	210.3
Agalawatta	5843	11.44	22.06	0	2.1	13.4	443.8
Water Level	5843	118.70	111.97	62.71	79.53	124.22	1355.63

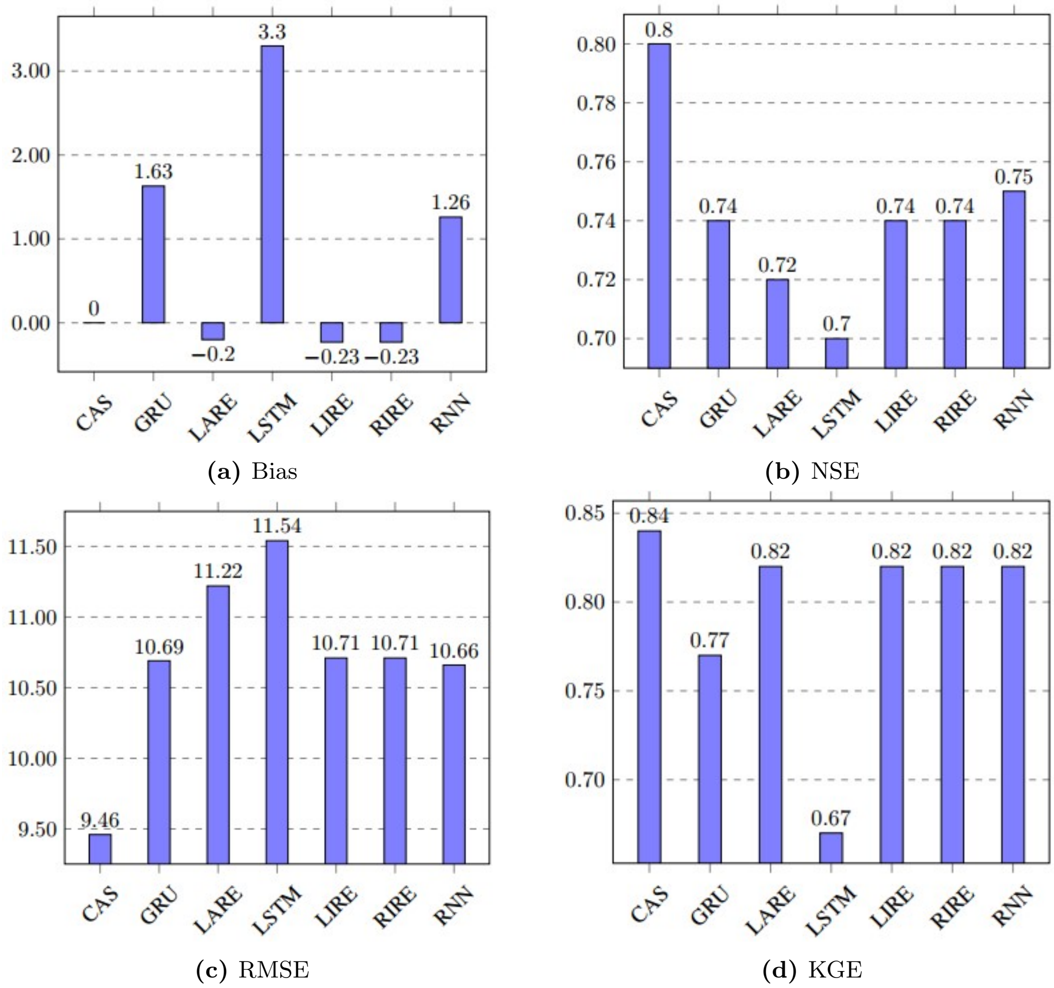
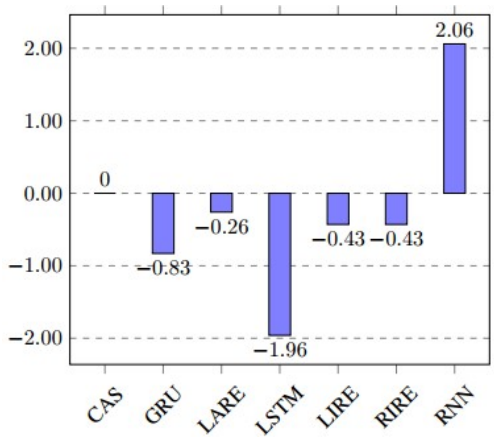
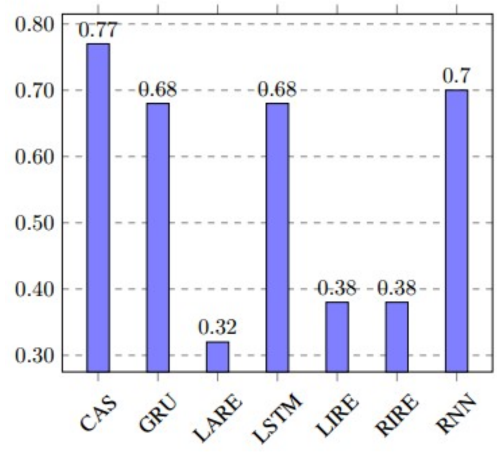


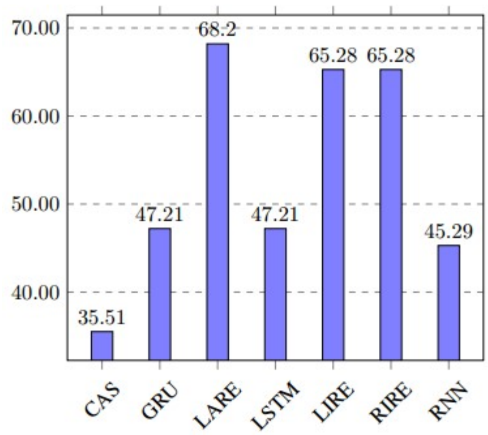
Figure A.1: Evaluation parameters results of Monobe River(a) Bias, (b) Nash-Sutcliffe efficiency (NSE), (c) Root Mean Square Error (RMSE), (d) Kling- Gupta Efficiency (KGE).



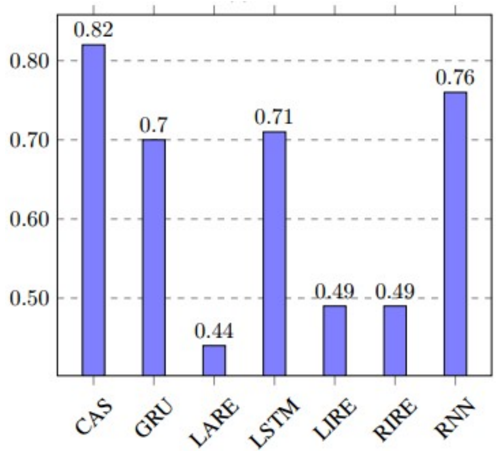
(a) Bias



(b) NSE



(c) RMSE



(d) KGE

Figure A.2: Evaluation parameters results of Niyodo River; (a) Bias, (b) Nash-Sutcliffe efficiency (NSE), (c) Root Mean Square Error (RMSE), (d) Kling- Gupta Efficiency (KGE).

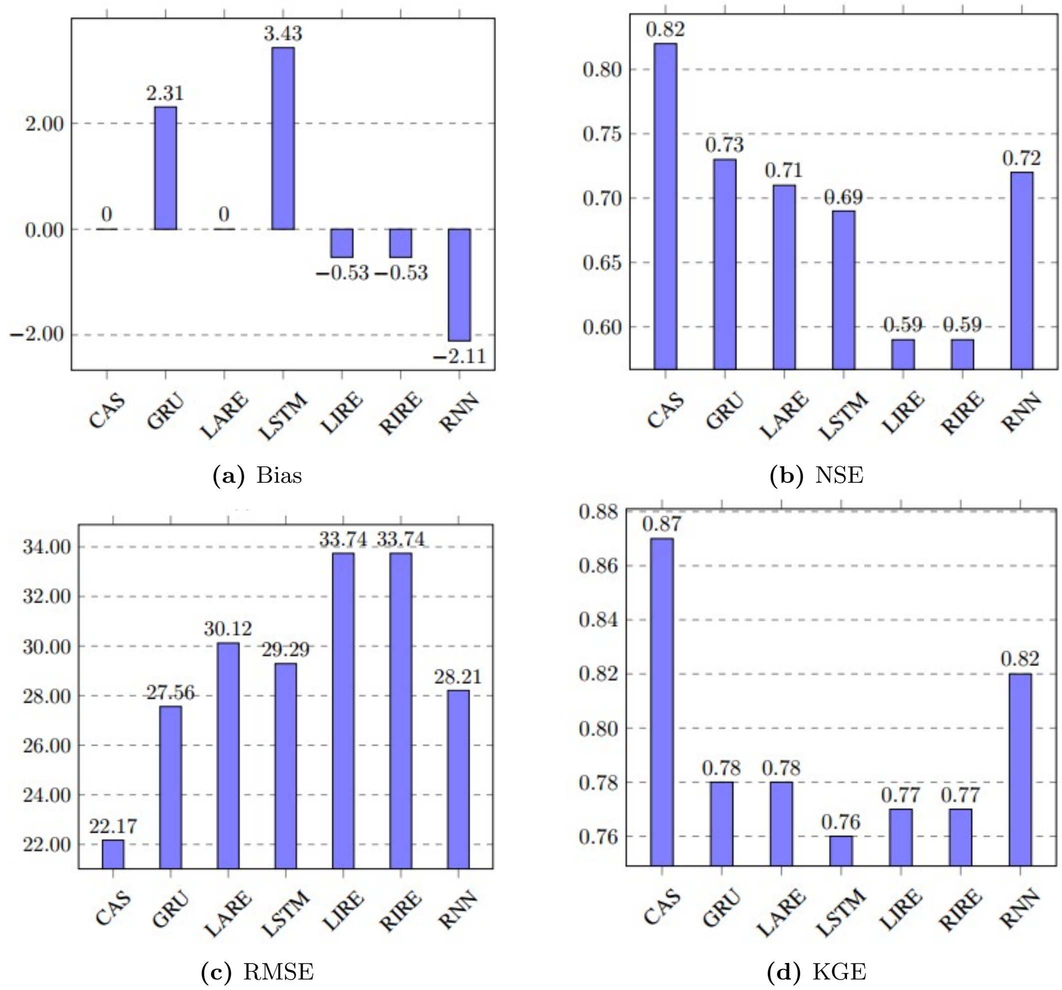


Figure A.3: Evaluation parameters results of Thu Bon River; (a) Bias, (b) Nash-Sutcliffe efficiency (NSE), (c) Root Mean Square Error (RMSE), (d) Kling- Gupta Efficiency (KGE).

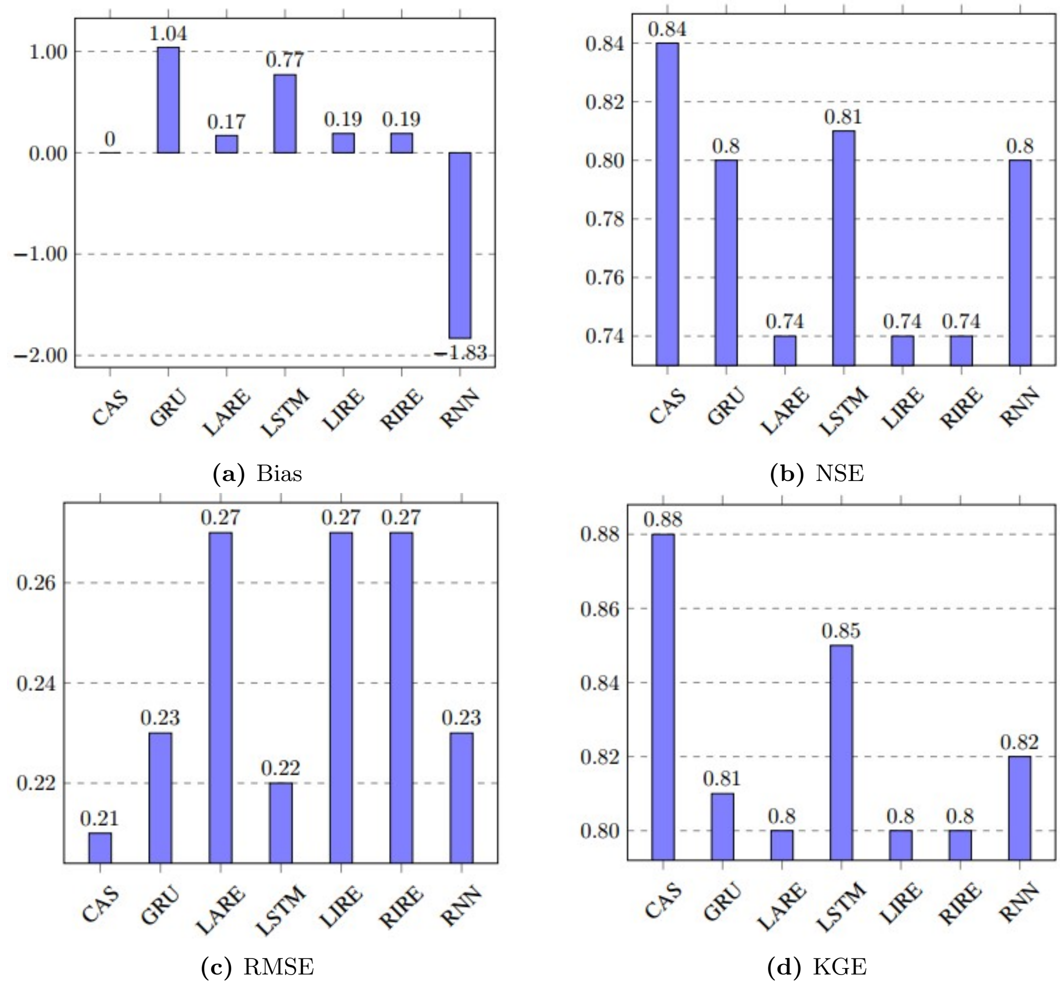
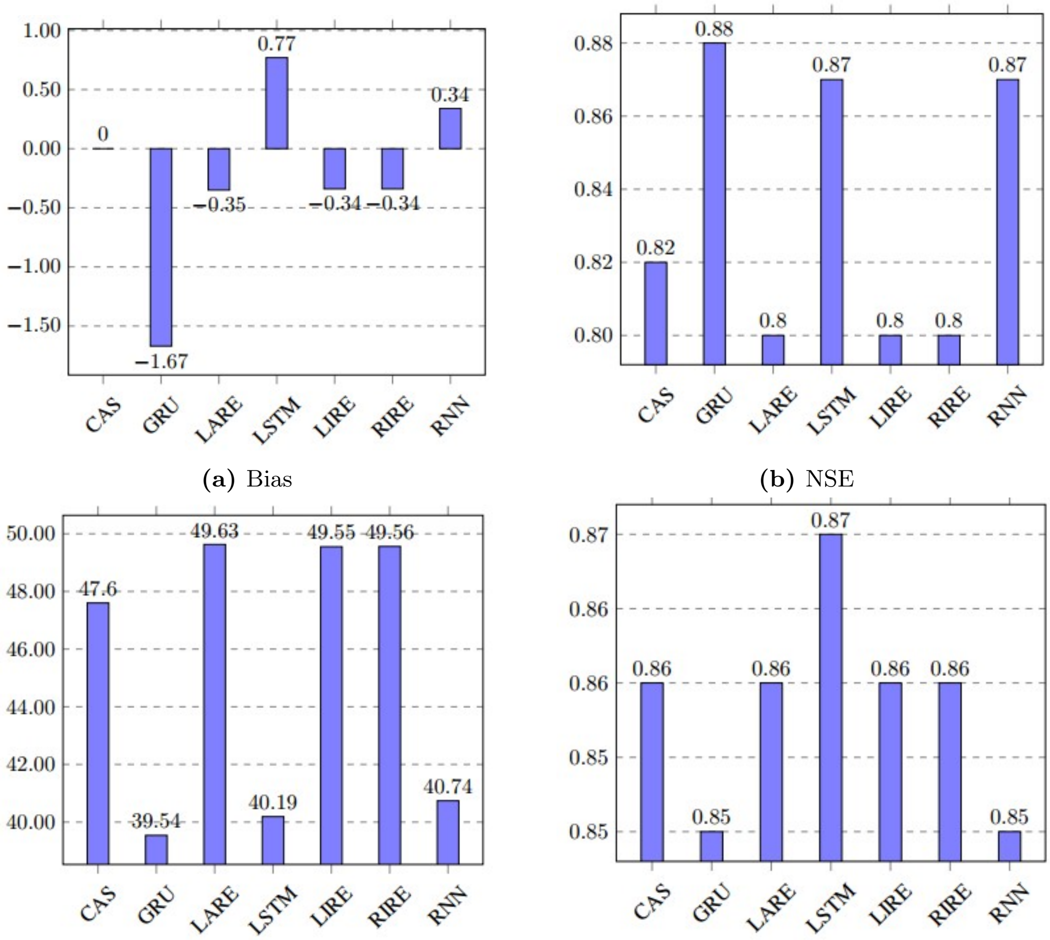


Figure A.4: Evaluation parameters results of Kelani River(a) Bias, (b) Nash-Sutcliffe efficiency (NSE), (c) Root Mean Square Error (RMSE), (d) Kling- Gupta Efficiency (KGE).



(a) Bias

(b) NSE

(c) RMSE

(d) KGE

Figure A.5: Evaluation parameters results of Kalu River(a) Bias, (b) Nash-Sutcliffe efficiency (NSE), (c) Root Mean Square Error (RMSE), (d) Kling- Gupta Efficiency (KGE).