

Research Article

Integrating ANFIS and Qt Framework to Develop a Mobile-Based Typhoon Rainfall Forecasting System

Shiu-Shin Lin ¹, Kai-Yang Zhu,¹ Jun-Yuan Wang,² and Ying-Po Liao³

¹Chung Yuan Christian University, Taoyuan, Taiwan

²Chuan Yun Engineering Technology Consultant Co., Ltd., Taoyuan, Taiwan

³Microsoft Corporation, Washington, USA

Correspondence should be addressed to Shiu-Shin Lin; linxx@cycu.edu.tw

Received 22 September 2021; Revised 21 February 2022; Accepted 18 April 2022; Published 14 May 2022

Academic Editor: Andrej Hrovat

Copyright © 2022 Shiu-Shin Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine learning methods such as Adaptive Network-Based Fuzzy Inference System (ANFIS) have been widely employed in intelligent urban storm water disaster warning for the purpose of smart city. However, there exists lack of research proposed for applying ANFIS and mobile application (App) to reach the purpose of smart city. In order to accomplish the goal, the study integrates ANFIS and Qt Framework to develop a Typhoon Rainfall Forecasting System to real-time typhoon rainfall forecast via a mobile device. The Service is first built by applying cluster analysis to typhoon data (Tamsui Weather Station of Taiwan) during June 1967 and November 2020 to classify the data into four groups and then applying the ANFIS to construct the Service with data in each group. The fuzzy rule of ANFIS is established by grid partition method. Both the Service and App employ Qt Framework as the cross-operating development tool, and the App is transformed to a smart mobile device App of different platforms. The simulated results show the following: (1) Taking the example of typhoon Nakri in group 1, the lowest root mean square error (7.898 mm) and the lowest computation time (178 sec) were obtained for training with 1000 steps and three membership functions. (2) Using the optimal parameters of the typhoon belonging to that group can obtain better prediction results. The developed typhoon rainfall forecasting system App in the supplementary information demonstrates that the user can use the smart mobile device for real-time typhoon forecasting at the most three hours ahead easily.

1. Introduction

Taiwan is located in the junction of the temperate and subtropical zone and is often attacked by typhoons from May to November every year. According to the statistical data of the Central Weather Bureau (CWB), Taiwan, it suffers from about 3.5 typhoons annually, with a rolling topography, a lot of mountainous regions, and short but rapid rivers; the heavy rainfall during typhoons sometimes causes severe flood disasters, which induce debris scouring in mountainous areas, leading to such major disasters as landslide and

debris flow. Typhoon Herb in 1996 was the best example; it carried the record-breaking torrential rain which caused a major disaster to the whole Taiwan. It is vital to predict typhoon rainfall in real time so that the government could mitigate the disasters even avoid them.

The typhoon rainfall forecasting can be divided in two methods: physical numerical method and statistical method [1–6]. As the physical numerical method employs numerical methods to solve complex fluid dynamic nonlinear equations. It requires more resources and equipments than the statistical model. In contrast, the statistical model, an

approach learning from data, more easily implemented than physical numerical method, analyzes the typhoon rainfall and related variables, and statistically learns the overall data from a lot of historical data to establish the relationship of forecasts.

The typhoon rainfall mechanism is a highly nonlinear process. Over the past 20 years, research related to typhoon rainfall forecasts has moved towards the field of artificial intelligence (AI). Researchers have attempted to address the limitations of traditional statistical methods by using machine learning algorithms and computational resources. AI-liked models such as fuzzy theory (FT), artificial neural network (ANN), and group method of data handling (GMDH) [4, 7–11], which have been proposed in recent years, can be used for forecasting typhoon intensity, tracks and rainfall [12–19]. Among many AI-liked methods, the ANFIS have been widely used in the field of hydrometeorology. In [4, 20–23], it combined a statistical analysis with AI theory to develop a composite real-time forecasting model. By combining the cluster analysis (CA) with ANN and ANFIS, the typhoon rainfall forecasting performance is enhanced.

In the past 10 years, the concepts of smart city and smart storm disaster prevention have been proposed that attempted to integrate cloud computing, Internet of Things (IoT), and artificial intelligence to reach the objective. It will make people’s living more convenient and decrease the impacts on urban storm disasters. The first of all for intelligent disaster prevention is to forecast the rainfall in leading time, and the second is to integrate the rainfall forecasting and the mobile devices (e.g., mobile phone) that people use in their daily life into a single mobile App. Only then can we achieve this goal. Previous studies have shown that using machine learning methods (e.g., ANFIS) to forecast typhoon rainfall can reduce the complex and time-consuming calculation of physical patterns. It can significantly reduce the computation time and make it easier to achieve the goal of real-time forecasting. If the mobile App can be used to forecast typhoons rainfall anytime and anywhere, the possibility of achieving the purpose of intelligent storm disaster prevention will be greatly enhanced. However, the related studies were rarely seen in the past.

As mentioned above, in order to achieve the purpose of intelligent rainfall prevention, this study develops and integrates real-time typhoon and rainfall forecasting and cross-platform mobile app based on ANFIS. The mobile devices are able to forecast the typhoon rainfall anytime and anywhere so that the purpose of urban rainfall mitigation can be achieved. This study employs the ANFIS with the CA theory to develop a Typhoon Rainfall Forecast System (TRF). The optimum parameters of TRF are first established by applying CA for typhoon track to group the data and then optimize the parameters of ANFIS based on the classified groups, respectively. Finally, rules and parameters of TRF are stored into database for the typhoon rainfall forecast being more instant. With the portability and handiness of smart mobile devices, the typhoon rainfall simulation can be performed by wireless network connection instantly; the typhoon rainfall information in the next a few hours is esti-

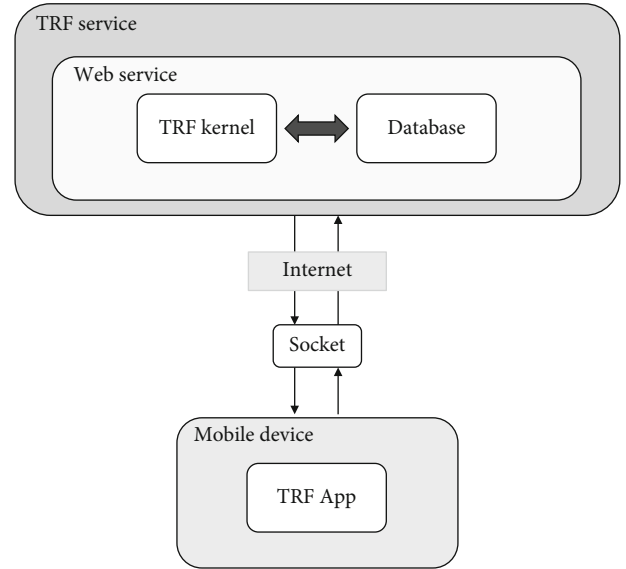


FIGURE 1: Architecture of the TRF system.

mated and displayed by the smart mobile device to provide a real-time forecasting effect, so as to implement real-time decisions on disaster prevention.

2. Typhoon Rainfall Forecast System Structure

The TRF system developed in this study is shown in Figure 1; it can be divided into TRF Service (Web Service) and TRF App. The TRF Service executes the communication between TRF Kernel and Database. It mainly handles the requests and corresponding operations from the client end (TRF App). As the requests sending from the TRF App, they are transmitted to the TRF Service by the Internet Protocol via the socket. As the socket being established through the TRF App and Web Service, data transmission and operations are completed by socket, and then the simulation result carried out by TRF Kernel is demonstrated on the mobile device (TRF App).

The TRF Kernel, real-time typhoon rainfall forecast model, simulating and forecasting typhoon rainfall, is an ANFIS-based computation engine. All the rules and parameters of TRF Kernel are pretrained and tested using the historical typhoon data. Afterwards, those of them are then stored in the Database. By selecting the typhoon event parameters from the Database directly, the user can implement a real-time forecast. The Database provides the TRF Kernel with the optimal model parameters which user inquiry to forecasting the typhoon rainfall. The Web Service is the bridge for communicating the TRF App with the TRF Kernel and Database.

After the server platform of TRF was built, an App of front end user interface was designed, called TRF App, developing to run on iOS, Android, and Windows operating systems. The smart mobile device performs a real-time forecast of typhoon rainfall and data transmission online, and the result is demonstrated at the front end.

3. Comparing Development Tool and Platform for Smart Mobile Device App

For the development of the mobile device App, as there are multiple brands of mobile devices, the operating systems are different, and the development modes and development program languages are also completely different. This leads to difficulties in developing an App [24, 25]. According to the mobile devices on the market and most users, the major mobile device operating systems are iOS, Android, and Windows operating systems. To develop an App for these three operating systems all at the same time, people tend to use their own development tool chains, so the development takes a lot of time and not efficient. The methods to develop Apps for the three operating systems are described below:

- (1) Native App development: The client program is developed generally by Objective-C, Java, C, C++, and C# program languages, and their own development tools are used. The system developers should know these programming languages to develop Apps using the iOS, Android, and Windows operating systems. The advantage is that the native operating system has high supportability, and as the C-like language is used, the execution efficiency is better than other languages [26]
- (2) Web App development: The client program is developed generally by HTML/XHTML/HTML5 + CSS + JavaScript + PHP web technologies and is executed by the mobile browser on the smart mobile device, which is a sort of Web. The advantage is that this development uses the same programming languages as the previous Web development languages which are universally acquainted by the developers
- (3) Hybrid App development: The same as Web App development mode, the PhoneGap compiling software framework tool enables the program at the user end to interact with the mobile device hardware. It can be done by using a partial native program and finishing the final program by wrapping it with a shell for Native App. The advantage is a lower development threshold; the workable App for iOS, Android, and Windows operating systems can be developed as long as a set of the same program codes is written and an amount of energy and time in the development process are reduced. However, the defect is that some settings are slightly different or unsupported [26, 27].

Based on the three development modes mentioned above, this study considered the issues of cross-platform and efficiency of execution and used Qt Framework as the TRF App development language and tool. The Qt Framework is the C++ library of messaging passing and event handling mechanisms between different classes (for example, Window class); the user can implement the App by inheriting the classes provided by the Qt Framework and implementing the message passing and event handling

mechanisms. As the Qt Framework is all developed by C++ and translated by Qt-Creator conversion technology into a native code App for smart mobile devices of different platforms, it has higher system compatibility and effectiveness.

As shown in Figure 2 of the Qt Framework class diagram, the QObject is the basic class of all classes. It defines the common attributes and methods of all Qt classes. The QObject event handling uses Qevent() related function; the event handling procedure will be triggered by the exec() method of QApplication object derived from QObject. All the classes inherited from QObject contain this method to event handling. For example, the functions of the dependent event are derived from Qevent(), e.g., sendEvent() and postEvent(). Several important classes in the Qt Framework are inherited from QObject, such as QApplication, QGraphicsScene, QGraphicsView, QGraphicsItem, QLayout, and QWidget. Among them, QApplication is derived from QApplication; the Scene is derived from QGraphicsScene; the View is derived from QGraphicsView; the Item is derived from QGraphicsItem; and the QBoxLayout, QHBoxLayout, and object unit QWidget are derived from QLayout.

The QWidget is the abstract class of the most fundamental components for Qt window development; all types of GUI components in Qt, such as button, label and toolbar, are derived from QWidget, and the QWidget itself is the subclass of QObject. The Widget is in charge of receiving such other events as mouse and keyboard from window system, which are displayed on the screen. Each GUI component is an entity of Widget class, the Widget can be used as a container, containing other Widget objects. All QWidget objects are statically connected to communicate components for event handling.

In terms of window event handling in Qt, the framework is processed by Graphics View. The Graphics View is a Model View Controller (MVC) model. In the Graphics View, the Scene stores graph data, which are displayed in multiple forms through View; the Item of each graph is controlled individually. Many Items of different bases have been built in Qt, new Items can be defined if necessary, and they can be grouped into single one object (e.g., QGraphicsItem QtframeworkGroup), if necessary.

As shown in Figure 3, the QGraphicsItemQtframeworkGroup can contain several Scenes, each Scene can contain several Views, and each View can contain many Items, but each Item comprises Widget, so as to compose a huge QGraphicsItem Qt frameworkGroup. A Group is composed of QGraphicsItemQtframeworkGroup for event handling.

All the Widgets in QGraphicsItem QtframeworkGroup have their own jobs. The Widgets with different functions are combined together, so as to achieve an event with complete function. The handling procedures of different events can be managed conveniently by QGraphicsItemQtframeworkGroup, and the function in QApplication clicks events and quits events.

The event handling refers to the user and window message passing mechanism; when the exec() of QApplication is executed through window, the App enters event loop to

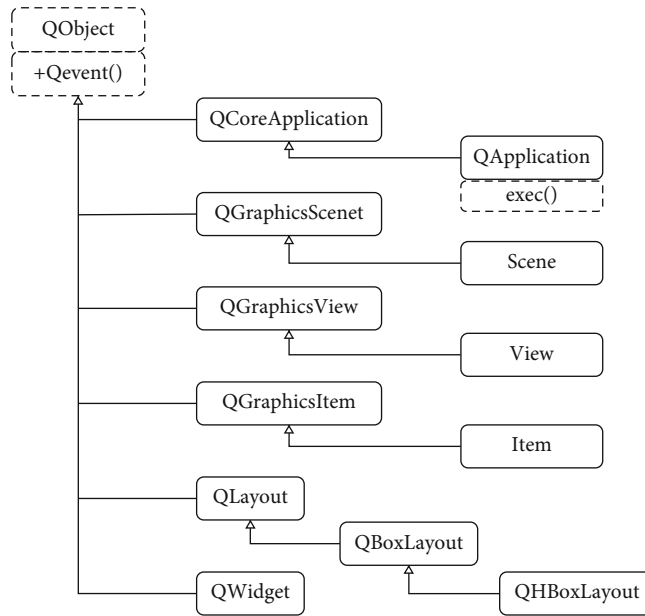


FIGURE 2: Class diagram of Qt Framework.

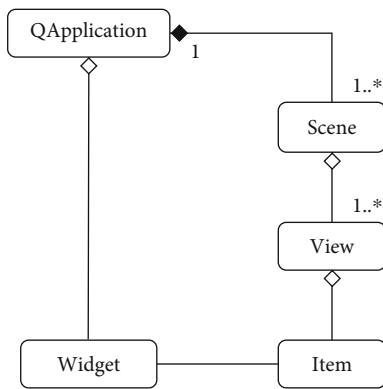


FIGURE 3: QGraphicsItemQtframeworkGroup.

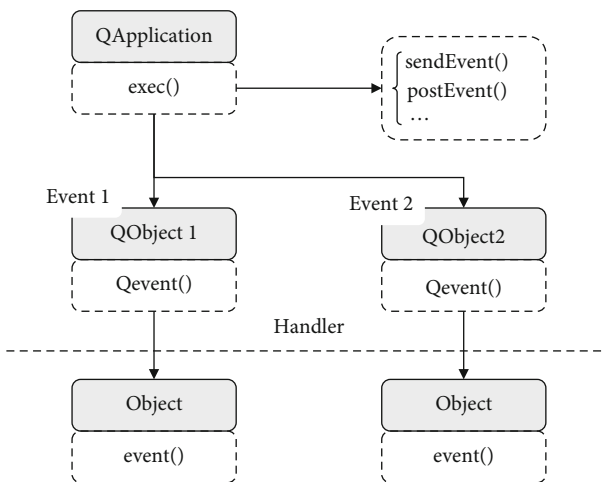


FIGURE 4: Event Handlers of QApplication.

process the event of App. The source of event is generally the window system, e.g., the user’s mouse event or keyboard event; it also can be the Qt App event itself. For example, the events of timer and network data are generated by the system individually and can be user-defined events, which are sent through `sendEvent()` or `postEvent()` of `QApplication`. The `QEvent` established by Qt itself is the basic class of all events; the `event()` function in `QObject` subclass transfers events; the `event()` itself does not handle events directly, but conveys events, when the event trigger message is sent to the corresponding Object to handle the event, e.g., Event Handler.

Each Event Handler only handles its own events, the events are independent of each other, but different objects may be used in different events. To avoid conflict, the event handling is generally asynchronous. The Qt Framework employs an event queue to process events. A new event is basically arranged at the tail of the queue. As the previous event is handled completely, the next event is the extracted from the front of the queue for further handling. The events of Qt can be synchronous if necessary, and the events can be filtered by event filter.

As shown in Figure 4, when a window (`QApplication`) generates an event, it is transferred to the `QObject` to receive information. Then, the handler Object of the event processes it. As stated above, each event is basically asynchronous, so the events are generally handled by scheduling. The `QApplication` object sends the message of Event 2 to `QObject2` to process the Event 2 until Event 1 is finished. The Object which completes event handling may comprise many `QWidget`s.

4. System Design and Implementation

4.1. System Design. The TRF system provides the cases of historical typhoon query, typhoon rainfall real-time

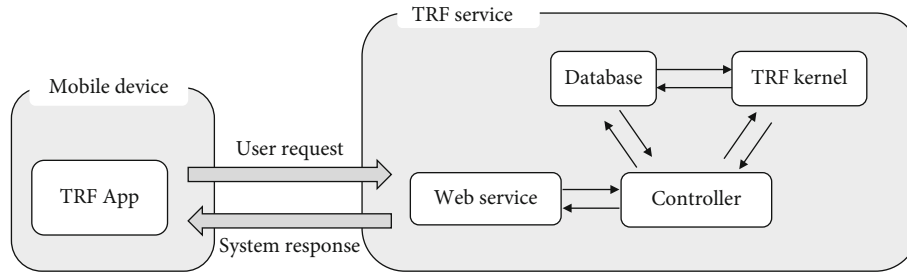


FIGURE 5: Architecture of TRF Service.

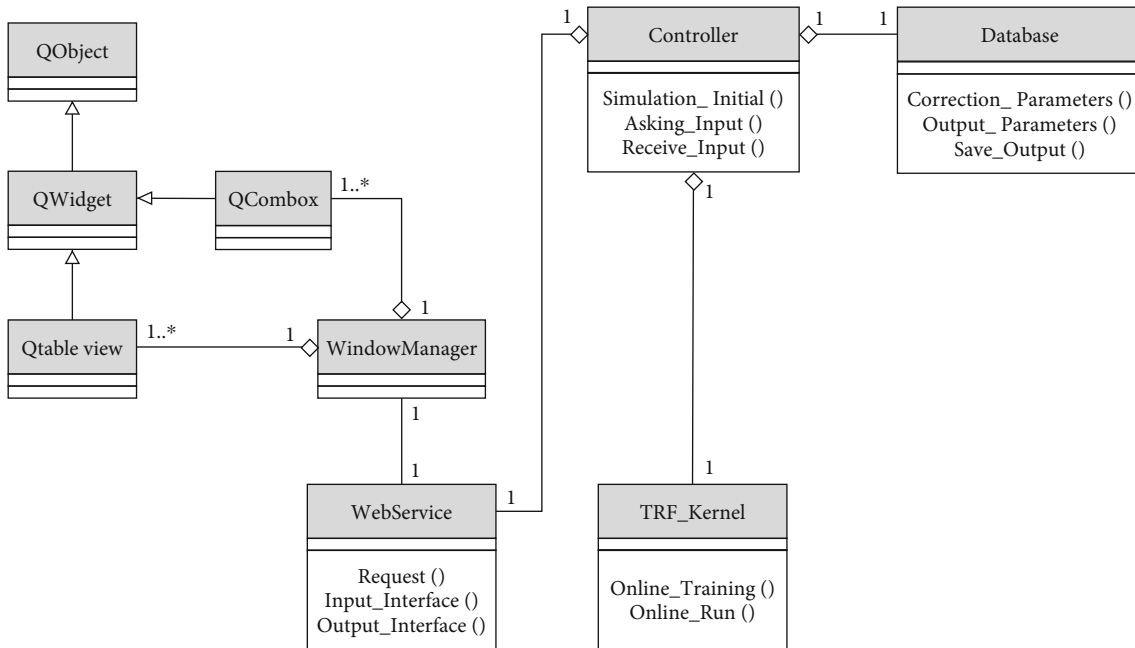


FIGURE 6: Class diagram of TRF.

simulation, and query of historical simulation data so that the users (e.g., hydraulicians and government disaster prevention bodies) can perform real-time simulation of typhoon rainfall forecast through the smart mobile device App.

Figure 5 is the architecture of TRF Service, including Web Service, Controller, TRF Kernel, and Database components. The user in the TRF App makes a request to the server. When the Web Service receives the user request, relevant information is then sent to the Controller component. The Controller component can execute the corresponding message handling according to User request. For example, when the user would like to execute hourly typhoon rainfall simulation, the trained typhoon parameters are selected from the Database by the Controller, and then the Controller calls the TRF Kernel component to calculate hourly typhoon rainfall simulation. After the TRF Kernel finishes the calculations, the hourly typhoon rainfall simulated result is sent to the Database to store it, and then the Controller calls the Web Service to respond the simulated result to the User.

Figure 6 shows the class diagram of TRF. The window is demonstrated by WindowManager, and all the window components (e.g., QComboBox and QtableView) are inherited from Qt Framework. The window message processing is implemented according to Qt Framework message processing mechanism; the Controller is in charge of controlling the communication with WebService, database, and TRF Kernel. Among them, the WebService handles the data transmission through Request(), Input_Interface(), and Output_Interface(); TRF_Kernel is the kernel of simulation. The parameters of the kernel are established by Online_Training(); the online real-time simulation is performed by Online_Run(); the database stores data. The model parameters are stored by Correction_Parameters() and Output_Parameters(), and the simulation results are stored by Save_Output() and respond them for demonstration.

Figure 7 uses the sequence diagram of Unified Modeling Language (UML) to describe the interactive relationship of TRF components. The TRF Service is the PC Windows version designed by Qt-Creator; the TRF Kernel is mainly composed of Matlab programs; the Database is composed of

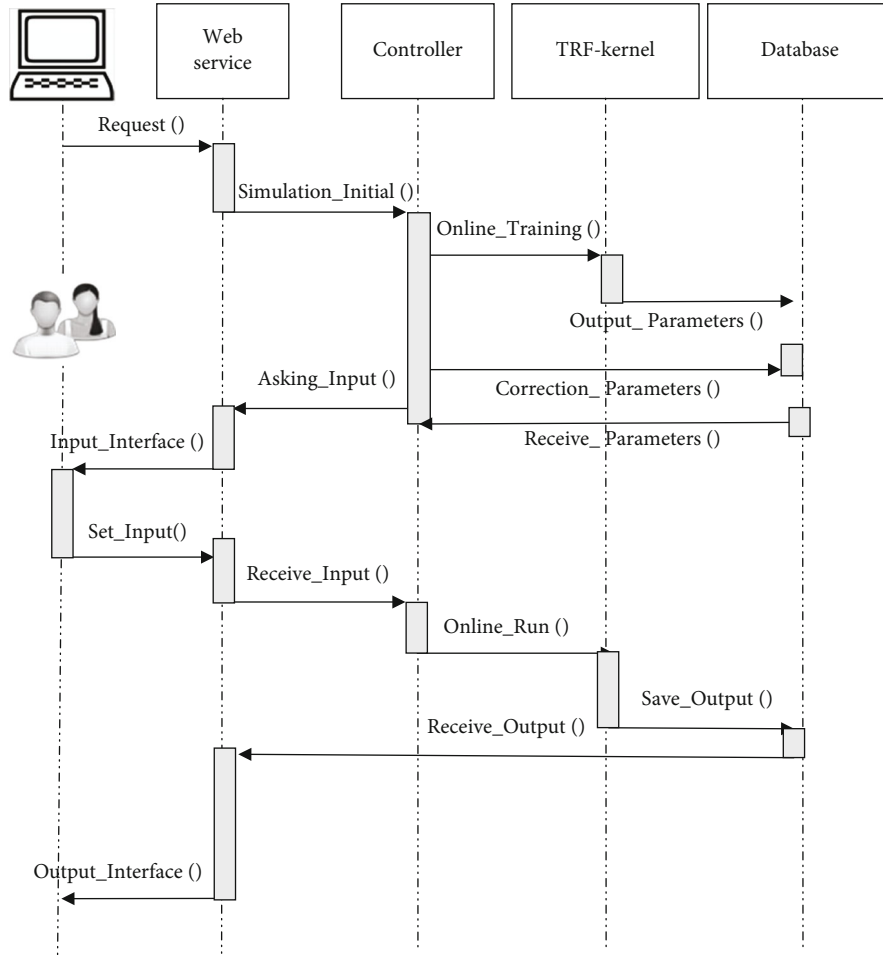


FIGURE 7: Sequence diagram of TRF Service.

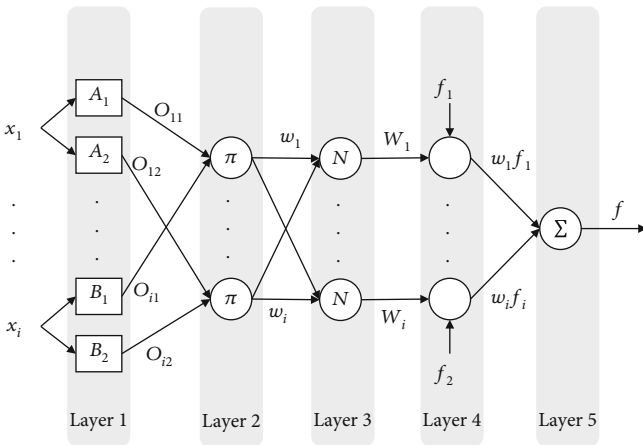


FIGURE 8: Architecture of adaptive neuro-fuzzy inference system.

SQLite and Service. The SQLite is slightly different from the other general databases. General ANSI SQL-92 syntax is workable, and it is unnecessary to build a database system, as long as the SQLite link library is compiled in the program.

The `Online_Training()` starts the ANFIS model parameters training. The `Output_Parameters()` stores the optimal parameters of TRF-Kernel (ANFIS based model) in the Database after training. If there are new typhoon data, the parameters are retrained by `Correction_Parameters()`, providing the mobile device the historical optimal parameters for rapid correction and simulation. After the ANFIS kernel is calibrated, related settings are inputted in succession, and the Controller calls `Online_Run()` to actuate the ANFIS module for simulation, i.e., the important kernel of real-time simulation of hourly typhoon rainfall. After simulation, the `Save_Output()` responds to the Service SQL component and back to the Controller and to the user to demonstrate the result.

4.2. System Implementation. Firstly, the TRF system classifies the historical data of typhoons into groups by cluster analysis [4], and then the TRF Kernel is built based on ANFIS with the corresponding classified data groups. Each grouped data are employed to train and validate the corresponding model to achieve the accuracy. The optimal parameters of each model built using the grouped data are stored in the Database. Finally, the typhoon rainfall can be predicted by applying the pre-stored parameters and the related variables.

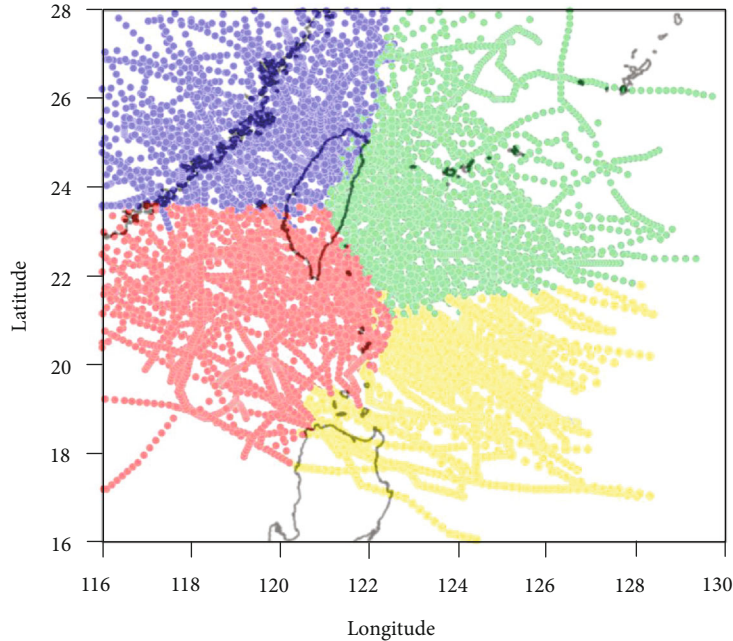


FIGURE 9: Results of spatial cluster analysis.

TABLE 1: RMSE and computation time of typhoon Nakri at different training steps and the number of MFs.

Training step	MF	RMSE (mm)	Time (sec)
100	2	8.334	1
300	2	8.332	5
1000	2	8.332	21
1000	3	7.898	178
1000	4	9.225	535
1000	5	11.734	3488

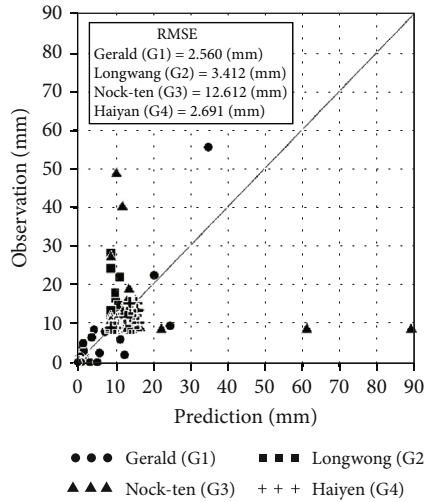
The TRF App is implemented by C++ language according to Figures 6 and 7; the Database applies SQLite for storing historical typhoon events, parameters of TRF Kernel and prior simulation results.

4.2.1. Spatial Cluster Analysis. In this study, the typhoon data were spatially clustered using Ward's clustering and k-means clustering. The two-stage clustering method can compensate for the disadvantages of hierarchical clustering and nonhierarchical clustering methods that are not easy to optimize the global clustering. The variables used are the longitude and latitude of the typhoon center of the training data [28, 29].

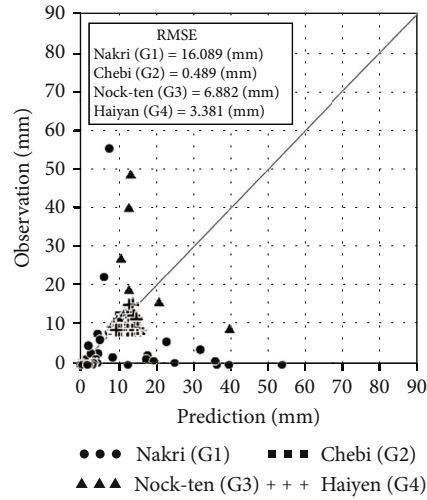
4.2.2. ANFIS. The ANFIS, employed as the TRF Kernel, is a neural network architecture based on fuzzy inference system [30] shown in Figure 8; it has the ability of fuzzy inference for dealing with uncertainty and imprecision, and it is able to organize itself, to optimize model parameters automatically. The IF-THEN rule is the first-order Sugeno fuzzy model, divided into five layers. Layer 1 to Layer 5 are the input layer, rule layer, normalization layer, result inference

layer, and output layer, respectively [4, 7, 8, 10, 21, 22]. The main concept is to convert the input value (x_1 to x_i) to a fuzzy value through the transformation of the memberships function (MFs). Then, calculations are made via the parameters (w_1 to w_i and W_1 to W_i) of each layer. Finally, the output (f) is calculated by the first-order Sugeno fuzzy model. The common MFs are triangular, bell-shaped, trapezoidal, and bell-shaped distributions. Among them, the best accurate outputs simulated are triangular and bell-shaped widely employed in various fields. In this study, the MF used is the triangular distribution for simplicity.

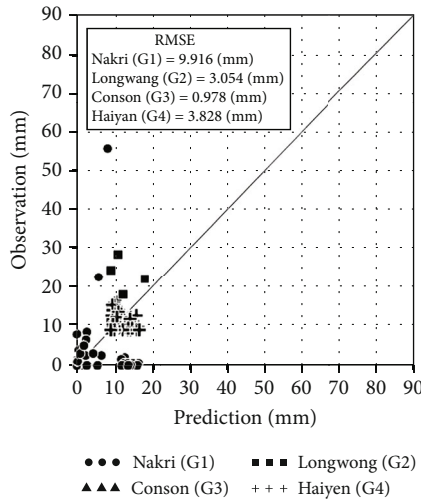
4.3. Results. This study took the Tamsui Weather Station ($121^\circ 26' 24''$ E, $25^\circ 09' 56''$ N, 19.0M, New Taipei City, Taiwan) as case study to demonstrate TRF system. The source of weather data of typhoon is the typhoon database of the Central Weather Bureau (<http://rdc28.cwb.gov.tw/>). All typhoon data of the Tamsui Weather Station during June 1967 and November 2020 (176 typhoon events) were used. They were first divided into four groups by the Subtractive Clustering Method [4]. Figure 9 shows that Group 1 to Group 4 (G1, G2, G3, and G4) had 42 typhoons, 92 typhoons, 30 typhoons and 12 typhoons, respectively. The classified typhoon data were partitioned by 4:1 for building data and testing data; the MF was a triangular distribution; the ANFIS parameter training and validation were performed. According to the study of Kan et al. [4], air pressure, temperature, humidity, wind speed, and predicted hourly typhoon rainfall were selected as input variables for future hourly rainfall forecasts. Table 1 shows the training results of the parameters of TRF Kernel. The Typhoon Nakri was chosen as an example with training 100 times, MF=2, with training 300 times, MF=2 with training 1,000 times, MF=2, training 1,000 times, MF=3, training 1,000 times, MF=4 and training 1,000 times, MF=5, respectively. It shows lower



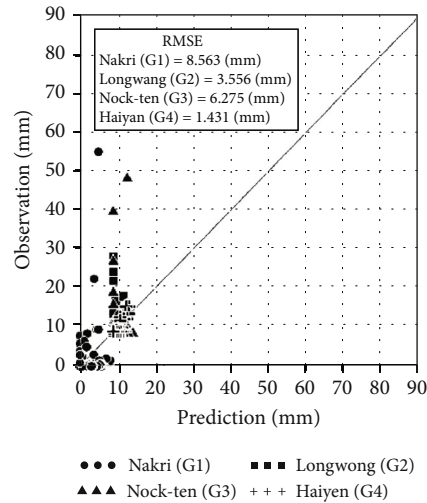
(a) Testing results of the optimal parameters for G1



(b) Testing results of the optimal parameters for G2



(c) Testing results of the optimal parameters for G3



(d) Testing results of the optimal parameters for G4

FIGURE 10: Testing results of optimal parameters of G1-G4.

RMSE and less execution time with training 1,000 times, $MF = 3$. However, the execution time will increase sharply, while the MF greater than 3. Because the purpose of this study is trying to find out what is the better combination between accuracy and computation time, 1,000 times of training and $MF = 3$ for model training of TRF Kernel is a better choice to achieve that.

A typhoon was selected randomly from the groups, and the optimal parameters of different groups were adopted for test, so as to analyze the influence of the beforehand classified typhoon on the test result. The findings show that if the optimal parameters of the group are applied to the typhoon event of the same group, a better trend is obtained, and the RMSE is lower than the other groups' test results. Figure 10(a) shows that four groups of typhoons were tested by using the optimal typhoon parameters of G1; the typhoons of G1 to G4 were Gerald, Longwang, Nock-ten, and Haiyan, respectively. In Figure 10, the forecasted and observed rainfall of Typhoon Gerald are highly correlated, and the RMSE is lower than the other three typhoons. The

forecasted rainfall of Typhoon Longwang is relatively underestimated; the forecasted one of Typhoon Nock-ten is over-rated at peak, and the RMSE is much larger than the other three groups. The forecasted rainfall of Typhoon Haiyan is close to the observed value. A phenomenon can be observed in Figure 10; if the optimal parameters of another group are applied, it shows large differences in the forecasted rainfall of typhoon to one group and slight differences in the forecasted rainfall of typhoon of the other two groups.

As stated above, preclustering typhoon events can obtain forecasted results relatively close to historical observations. This study performs a forecast of 1-3 hours ahead for Typhoon Nakri. Figure 11 shows the original forecasted and the one-three hours ahead forecasted results of typhoon Nakri. It is observed that the trends of the four forecasted results are close to the observations and the global peak of two hours ahead forecast is the closest to the observed value. However, the four forecasted results are similar at local peak; a forecasted result will not be completely larger than the other three.

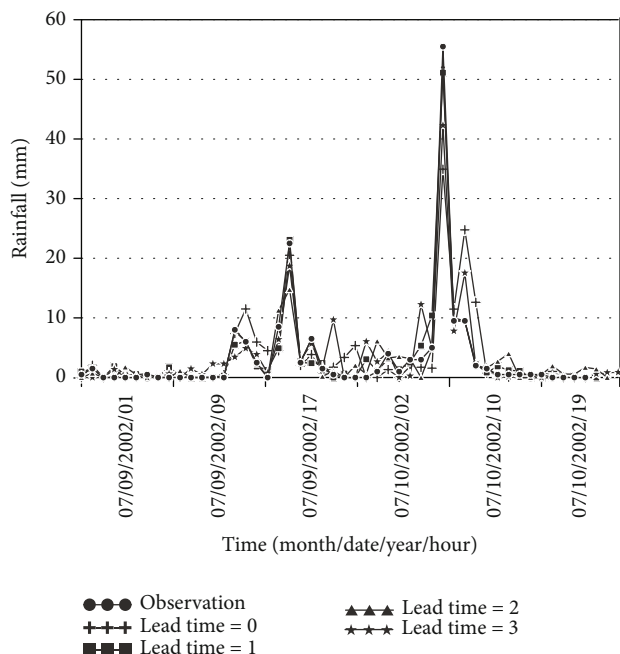


FIGURE 11: Predicted results for different lead times-taking typhoon Nakri for example.

5. Advantages and Limitations

The TRF system developed in this study integrates a cross-platform mobile device development tool (the Qt Framework) and the ANFIS model to forecast typhoon rainfall in real time. It attempts to further achieve intelligent rainstorm disaster prevention and smart city. The advantages and limitations are described briefly below:

- (1) The ANFIS can be a self-learning model and optimize the model parameters appropriately based on new data automatically. However, in the study, the ANFIS model still requires a semi-manual approach to update the model parameters
- (2) The Qt Framework and C++ programming language were used to develop cross-platform Native APP with better performance. It is better than other development tools for real-time rainfall forecasts

6. Conclusions and Recommendations

This study proposes a novel approach to integrates an AI-based model (the ANFIS model) and a device independent, cross-platform development framework (the Qt Framework) to build an operating systems independent, device independent typhoon rainfall forecasting system for real-time typhoon rainfall prediction. The proposed approach can be applied in advance to achieve the purpose of smart city and smart storm prevention. The historical typhoon data were used to build the TRF Service, Kernel (the ANFIS model), and then the TRF App.

First of all, the historical typhoon events were classified according to the cluster analysis results, the TRF Kernel was established and the typhoon group parameters were optimized with the typhoon data. Then, the Qt Framework and socket network connection mechanism were integrated with TRF to develop TRF Service and TRF App. According to the findings, the following conclusions are obtained:

- (1) The model parameters were set to take Typhoon Nakri as an example; training 100 times, MF = 2; training 300 times, MF = 2; training 1,000 times, MF = 2; and training 1,000 times, MF = 3 were employed to establish ANFIS parameters. The results show that the 1,000 times training and MF = 3 can achieve quite high accuracy. However, if the number of MFs is greater than 3, the computation time will increase significantly
- (2) According to the test results of optimal parameters of four groups, better forecasted results can be obtained by selecting the optimal parameters of the same group. Therefore, the TRF Service provided the optimal parameters of four groups, and appropriate parameters were selected according to typhoon information to perform online real-time forecasts. The native App developed in this study can be installed in the mobile devices with Android, iOS, and Windows operating systems to implement the online real-time typhoon forecasting
- (3) This study employed Qt for cross-operating platform development, the C++ was adopted for implementation, and the program code was converted by the Qt-Creator into native code to avoid the compatibility problem of different operating platforms and to increase the operating efficiency
- (4) This study employed cluster analysis and ANFIS to implement TRF Kernel, including the optimal parameters of groups established beforehand and the one-three hours ahead typhoon rainfall forecast information. Hence, the urban rainstorm disaster prevention is much more timely
- (5) The architecture of the TRF system is not a fully cloud-based system. It cannot be easy to co-operate with other well-known cloud service, such as Google, Microsoft, and Amazon

Data Availability

The source of weather data of typhoon is the typhoon database of the Central Weather Bureau, Taiwan (<https://rdc28.cwb.gov.tw/>).

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

The authors would like to thank the financial support from the Ministry of Science and Technology, Taiwan, under grant number MOST 109-2625-M-033-00-.

Supplementary Materials

Figure S1 to S5 are the demonstrations of TRF system. The user should select a group of typhoon parameters before simulation first, and then the user manually inputs the typhoon variables for rainfall forecast. The simulated typhoon rainfall could be obtained and displayed them on the UI. Figure S1 demonstrates the main page of TRF App. The “typhoon rainfall forecast” function button is the core function for the user to simulate typhoon rainfall instantly; the “historical typhoon data query” button provides the historical typhoon record of Tamsui Weather Station for user’s reference; the “forecast result query” button allows the user to inquire the completed previous simulated results, which are stored in the Database, and it is unnecessary to reperform the same complete simulation. In the “typhoon rainfall forecast” button, the user is required to select the typhoon parameters, the system provides four groups of parameters for simulation, including the optimal parameters of G1 to G4 (Figure S2). The user can select parameters according to the typhoon track released by the CWB, Taiwan, and the simulated results of the parameters of the corresponding group can be validated. After selection, the interface for inputting typhoon variables is entered (Figure S3); after the user’s manual input, the “forecast” is selected to display the simulated result (Figure S4). After completed simulation, in the “forecast data query” button (Figure S5), the user can inquire about the results by selecting the recording time, and then the “inquire” is executed; the prior simulated results will be displayed on the TRF App. (*Supplementary Materials*)

References

- [1] P. O. Bojang, T. C. Yang, Q. B. Pham, and P. S. Yu, “Linking singular spectrum analysis and machine learning for monthly rainfall forecasting,” *Applied Sciences*, vol. 10, no. 9, p. 3224, 2020.
- [2] C. C. Wei and T. H. Huang, “Modular neural networks with fully convolutional networks for typhoon-induced short-term rainfall predictions,” *Sensors*, vol. 21, no. 12, pp. 4200–4200, 2021.
- [3] I. J. Wen and C. H. Yang, “Using fuzzy time-series prediction for forecasting rainfall in Taiwan,” *International Journal of Organizational Innovation*, vol. 13, no. 1, pp. 235–254, 2020.
- [4] P. H. Kan, S. S. Lin, Y. L. Hu, and J. H. Sung, “Integrating non-linear principal component analysis and GA-fuzzy model to develop a typhoon rainfall forecasting model,” *Taiwan Water Conservancy*, vol. 65, no. 4, pp. 42–53, 2017.
- [5] C. C. Wei and C. C. Hsu, “Real-time rainfall forecasts based on radar reflectivity during typhoons: case study in southeastern Taiwan,” *Sensors (Basel, Switzerland)*, vol. 21, no. 4, 2021.
- [6] Q. Zhao, Y. Liu, X. Ma, W. Yao, Y. Yao, and X. Li, “An improved rainfall forecasting model based on GNSS observations,” *IEEE Transactions on Geoscience & Remote Sensing*, vol. 58, no. 7, pp. 4891–4900, 2020.
- [7] A. Kaboli, S. Hr, A. Al Hinai, A. H. Al-Badi, Y. Charabi, and A. Al Saifi, “Prediction of metallic conductor voltage owing to electromagnetic coupling via a hybrid ANFIS and backtracking search algorithm,” *Energies*, vol. 12, no. 19, p. 3651, 2019.
- [8] F. J. Chang and Y. Y. Hwang, “A self-organization algorithm for real-time flood forecast,” *Hydrological Processes*, vol. 13, no. 2, pp. 123–138, 1999.
- [9] T. Jacob, U. A. Usman, S. Bemdoo, and A. A. Susan, “Short-term electrical energy consumption forecasting using GMDH-type neural network,” *Journal of Electrical and Electronic Engineering*, vol. 3, no. 3, p. 42, 2015.
- [10] A. Pourdaryaei, H. Mokhlis, H. A. Illias, S. H. A. Kaboli, and S. Ahmad, “Short-term electricity price forecasting via hybrid backtracking search algorithm and ANFIS approach,” *IEEE Access*, vol. 7, pp. 77674–77691, 2019.
- [11] G. Wang, D. Wang, W. Han, and J. Yin, “Typhoon cloud system identification and forecasting using the Feng-Yun 4A/Advanced Geosynchronous Radiation Imager based on an improved fuzzy clustering and optical flow method,” *Advances in Meteorology*, vol. 2019, Article ID 5890794, 11 pages, 2019.
- [12] J. Devaraj, S. Ganesan, R. M. Elavarasan, and U. Subramaniam, “A novel deep learning based model for tropical intensity estimation and post-disaster management of hurricanes,” *Applied Sciences*, vol. 11, no. 9, pp. 4129–4129, 2021.
- [13] Y. Huang, X. Huang, H. Zhao, and J. Jin, “The influence of environmental factors on the intensity of typhoon Mirinae in the South China Sea and its forecast analysis,” *IOP Conference Series: Earth and Environmental Science*, vol. 658, no. 1, article 012019, 2021.
- [14] H. J. Kim, I. J. Moon, and M. Kim, “Statistical prediction of typhoon-induced accumulated rainfall over the Korean Peninsula based on storm and rainfall data,” *Meteorological Applications*, vol. 27, no. 1, pp. 1–18, 2020.
- [15] D. Li, Y. Liu, and C. Chen, “MSDM v1. 0: a machine learning model for precipitation nowcasting over eastern China using multisource data,” *Geoscientific Model Development*, vol. 14, no. 6, pp. 4019–4034, 2021.
- [16] E. K. Wang, F. Wang, S. Kumari, J. H. Yeh, and C. M. Chen, “Intelligent monitor for typhoon in IoT system of smart city,” *Journal of Supercomputing*, vol. 77, no. 3, pp. 3024–3043, 2021.
- [17] C. C. Wei and P. Y. Hsieh, “Estimation of hourly rainfall during typhoons using radar mosaic-based convolutional neural networks,” *Remote Sensing*, vol. 12, no. 5, p. 896, 2020.
- [18] S. Yuan, C. Wang, B. Mu, F. Zhou, and W. Duan, “Typhoon intensity forecasting based on LSTM using the rolling forecast method,” *Algorithms*, vol. 14, no. 3, pp. 83–83, 2021.
- [19] Y. Zhou, Y. Li, J. Jin et al., “Stepwise identification of influencing factors and prediction of typhoon precipitation in Anhui Province based on the Back propagation neural network model,” *Water*, vol. 13, no. 4, p. 550, 2021.
- [20] F. J. Chang and Y. T. Chang, “Adaptive neuro-fuzzy inference system for prediction of water level in reservoir,” *Advances in Water Resources*, vol. 29, no. 1, pp. 1–10, 2006.
- [21] F. J. Chang, Y. M. Chiang, M. J. Tsai, M. C. Shieh, K. L. Hsu, and S. Sorooshian, “Watershed rainfall forecasting using neuro-fuzzy networks with the assimilation of multi-sensor information,” *Journal of Hydrology*, vol. 508, pp. 374–384, 2014.

- [22] F. J. Chang, C. H. Chung, P. A. Chen, C. W. Liu, A. Coynel, and G. Vachaud, "Assessment of arsenic concentration in stream water using neuro fuzzy networks with factor analysis," *Science of the Total Environment*, vol. 494-495, pp. 202–210, 2014.
- [23] S. H. Chen, Y. H. Lin, L. C. Chang, and F. J. Chang, "The strategy of building a flood forecast model by neuro-fuzzy network," *Hydrological Processes*, vol. 20, no. 7, pp. 1525–1540, 2006.
- [24] M. Omar, M. M. Nawi, J. Jamil, A. Mohamad, and S. Kamaruddin, "Research design of mobile based decision support for early flood warning system," *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 14, no. 17, p. 130, 2020.
- [25] Y. Wang, W. Zhang, B. A. Engel et al., "A fast mobile early warning system for water quality emergency risk in ungauged river basins," *Environmental Modelling & Software*, vol. 73, pp. 76–89, 2015.
- [26] P. Nawrocki, K. Wrona, M. Marczak, and B. Sniezynski, "A comparison of native and cross-platform frameworks for mobile applications," *Computer*, vol. 54, no. 3, pp. 18–27, 2021.
- [27] C. M. Ferreira, M. J. Peixoto, P. A. Duarte et al., "An evaluation of cross-platform frameworks for multimedia mobile applications development," *IEEE Latin America Transactions*, vol. 16, no. 4, pp. 1206–1212, 2018.
- [28] D. C. Li, W. L. Dai, and W. T. Tseng, "A two-stage clustering method to analyze customer characteristics to build discriminative customer management: a case of textile manufacturing business," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7186–7191, 2011.
- [29] C. V. Trappey, A. J. Trappey, A. C. Chang, and A. Y. Huang, "Clustering analysis prioritization of automobile logistics services," *Industrial Management & Data Systems*, vol. 110, no. 5, pp. 731–743, 2010.
- [30] J. S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [31] J. S. Kim, A. Chen, J. Lee, I. J. Moon, and Y. I. Moon, "Statistical prediction of typhoon-induced rainfall over China using historical rainfall, tracks, and intensity of typhoon in the Western North Pacific," *Remote Sensing*, vol. 12, no. 24, pp. 4133–4133, 2020.