Enhancing River Flood Prediction in Early Warning Systems Using Fuzzy Logic-Based Learning

Rinta Kridalukmana, Dania Eridani*, Risma Septiana, Ike Pertiwi Windasari

Department of Computer Engineering, Diponegoro University, Semarang, Indonesia

Received 29 February 2024; received in revised form 24 May 2024; accepted 28 May 2024

DOI: https://doi.org/10.46604/ijeti.2024.13426

Abstract

Previous studies show that the fuzzy-based approach predicts incoming floods better than machine learning (ML). However, with numerous observation points, difficulties in manually determining fuzzy rules and membership values increase. This research proposes a novel fuzzy logic-based learning (FLBL) that embeds missing data imputations and a fuzzy rule optimization strategy to enhance ML performance while still benefiting from fuzzy theory. The simple moving average handles sensors' missing data. The logical mapping is used for fuzzification automation and fuzzy rule generation. The join function between the Szymkiewicz–Simpson coefficient similarity and max function is applied to optimize a fuzzy rules model. The case study uses observation data from three rivers traversing three districts in Semarang City. As a result, FLBL achieves 97.87% accuracy in predicting flood, outperforming the decision tree (96%) and the neural network (73.07%). This work is significant as a part of preventive flood-related disaster plans.

Keywords: flood prediction, flood prevention, flood situation awareness, fuzzy logic-based learning

1. Introduction

River flooding is considered one type of flood that can cause massive damage to society. This typical flood occurs when the water level exceeds the river's capacity, allowing the water to overflow into the surrounding occupied area. Some government agencies have developed a preventive flood-related disaster plan to mitigate the harmful effects of river floods, such as river maintenance and an early warning system (EWS) [1]. With the advancement of IoT technologies, providing an EWS solution has become a preference for improving people's situational awareness of incoming flood events [2-3].

In establishing a reliable IoT-based EWS architecture, critical requirements should be fulfilled, including a set of reliable sensory tools, good connectivity, reliable electric power sources, reasoning engines to infer the data from the sensors to generate early warning alarms, and apps to convey critical flood-related situations to related parties [4-6]. The architecture of an IoT-based EWS becomes more challenging when dealing with broader areas (e.g., inter-city) having many rivers traversing them. In this case, the rain in a particular area might cause river flooding in other areas even though no rain is detected. Moreover, there are many possibilities of impacted areas depending on the water levels of the traversing rivers and the land contour. Consequently, the performance of EWS's reasoning engine is at stake to infer incoming flood events, given the complexity of natural flood-related factors from areas of concern and data transmission problems coming from the IoT devices [7-10]. Therefore, a reliable reasoning engine for EWS is critically required to deliver flood-event alarms in critical times [11-12] and to impute missing data caused by connectivity problems.

* Corresponding author. E-mail address: dania@ce.undip.ac.id

English language proofreader: Yen-Chun Hsieh

In developing the reasoning engine for river flooding prediction, previous studies relied on three main approaches: rule-based reasoning (e.g., [3]), machine learning (ML) (e.g., decision tree [4], neural network [13-14]), and fuzzy-based approach (e.g., [15-18]). Some studies combine ML techniques with fuzzy theory, such as neuro-fuzzy [11, 19]. Rule-based reasoning may have higher reliability than ML techniques, particularly when data is insufficient to train as a learning model [15]. However, manually developing rules for complex situations is also a difficult process, which can be mitigated by ML techniques when good quality recorded data is available.

On the other hand, Lohani et al. [15] viewed the fuzzy-based technique as promising to improve real-time river flood prediction. During the early experiments, the fuzzy-based technique outperformed the ML technique in predicting flood events. However, determining fuzzy rules and membership values requires trial and error in complex scenarios [17]. Furthermore, some of the above studies also implemented missing data handlers, such as [13], which used the mean value of previously recorded data. Other techniques of missing data imputation are the rolling statistical technique, linear models, and regression imputation [20].

Even though the performance of methods proposed in previous studies had been evaluated in terms of accuracy, there are some hurdles, particularly in setting them up in an EWS for a wide area involving multiple clusters of observation points. Their performance in such settings has not yet been extensively examined. Moreover, this paper views optimizing fuzzy rule construction as an option to eliminate the trial-and-error process when determining rules and degree of membership. This option can be achieved by implementing the automatic learning capability from a dataset to generate fuzzy rules without losing the uncertainty model provided by the fuzzy and applying a rule optimization strategy. To fill these gaps, this study aims to develop a novel approach for multiple-clusters-based EWS to infer incoming flood events based on fuzzy logic-based learning (FLBL) and fuzzy rule optimization strategy, including missing value handler.

The rolling statistical technique is selected for its reliability and easy implementation to impute missing data compared to similar methods, such as linear models with autoregressive moving averages suitable for time series data. Autoregressive and moving-average models frequently express this association in error terms [21]. Furthermore, the fuzzy rule model is generated by the FLBL. Even though the FLBL has been introduced in different domains [22], none have been implemented in the hydrological field, especially in predicting incoming flood events. Moreover, the other FLBLs from previous studies were implemented in the Takagi-Sugeno (TS) [23] and Mamdani models [22]. This paper implements FLBL in a different setting based on fuzzy logic that uses logical mapping from input to output to generate fuzzy rules. The rules optimization strategy is also applied in this FLBL EWS system by finding similar rules using the Szymkiewicz–Simpson coefficient (SSC) and reducing them using the maximum function on their degree of membership values.

To sum up, the main contributions of this paper are as follows:

(1) A LoRa-based EWS architecture for flood-related data transmissions that supports multi-clusters-based observation points.
(2) The FLBL technique with optimization strategy is used to produce the fuzzy rules model to enhance incoming flood event alert predictions, including the simple moving average technique to impute the missing data.
(3) The proposed approach is demonstrated to prove the performance of the missing data handler and the fuzzy rules model.

The proposed method was showcased in three districts in Semarang City: Tugu, Ngaliyan, and West Semarang Districts. The flood event in those districts is affected by three traversing rivers: Plumbon, Bringin, and Silandak. The water level and rainfall rate sensors are two primary sensing tools utilized in this research. The experiments showed that the proposed method outperformed the baseline method in terms of accuracy in predicting flood events in selected areas. The average accuracy of this proposed method was 97.87%, while the accuracy of baseline methods using the decision tree algorithm and neural network was 96.83% and 73.07%, respectively.

2. Related Works

IoT-based flood detection for early warning has attracted researchers around the globe, and various methods of inference engines have been proposed, including flood warning strategies to raise citizen awareness [24-25]. Determining a threshold value of critical variables is the most popular method to trigger this warning. For example, Tedla et al. [26] and Gambini et al. [27] set thresholds on several variables, including rainfall rate threshold, antecedent precipitation index, and river water level. Those variables are combined with data from a satellite called Sentinel-1 Synthetic Aperture, citizen science, and stream flow data to simulate the flood extent.

As a result, predictions can be produced three days in advance, but the derived thresholds must be progressively adjusted based on new observation data. Furthermore, Sayyad et al. [3] used a rule-based method to produce early warning information for the control room. This method utilizes HC SR04 ultrasonic sensors to measure the river's water level and soil moisture sensors to detect electricity levels on wet soil. Those sensors were connected to an SG90 servo motor to control the dam gate through the ESP8266 Wi-Fi module. Even though the developed rules were reliable for the designated case, the proposed flood detection in this study was configured for a single spot where the dam gate is located, and there was no centralized mechanism to infer incoming flood events from multiple spots or clusters.

When multiple sensors are involved to collect flood-related data at an observation point, researchers prefer a machine-learning technique to develop a classifier to trigger flood-event alerts. For example, Vinothini and Jayanthy [4] proposed an IoT architecture to send data from temperature, humidity, and water level sensors to the cloud over an ESP8266 Wi-Fi module embedded in the PIC16f877a microcontroller. The water levels were measured at three stages involving conductive level sensors to examine conductivity between the ground and water. The first conductive level indicated ordinary water level, and the second represented above normal water level. Finally, the third conductive level was the hazardous water level state. The decision tree model installed in the cloud was used to infer incoming floods and achieved 99.6% accuracy during the laboratory experimental setup. Furthermore, other researchers used neural networks to infer flood status.

The two-class neural network was proposed by Abdullahi et al. [14] and applied in the cloud to support the inference module that obtains data over the Wi-Fi module from the miniaturized water level sensor. NodeMCU was utilized as the IoT hardware enabler that supports IEEE 802.11b/g/n. The connection to the cloud was conducted using HTTP protocol and REST API. The two-class neural network generated safe-unsafe labels to trigger incoming flood alerts. During the laboratory experimental test, the accuracy achieved 98.9%.

Anbarasan et al. [13] used a more sophisticated machine-learning approach that used the deep convolutional neural network to generate a learning model from IoT-based big data. The flood-related data consists of four variables: water flow, water level, rainfall rate, and humidity. The classifier's output indicates whether there is a chance of an incoming flood event. The learning model was trained using the National Weather Service, and as a result, this method achieved 93.23% accuracy in flood prediction.

In a research conducted by Prakash et al. [28], flood alert was developed by generating an ML model using extreme gradient boosting to estimate water discharge given changing geographic constraints. Seven types of hydro-meteorological data were collected using the IoT platform, including rainfall rate, river flow, river level, temperature, humidity, wind direction, and wind speed. The data were classified into four classes: very dangerous, dangerous, warning, and normal. The microcontroller Arduino UNO R3 collected data from the sensing unit, such as the water flow sensor YF-S201, the water level ultrasonic sensor HC-SR04, and the bucket rain gauge. The generated model was validated using a coefficient of determination (R^2) and relative error (RE) indicators and achieved 0.96 and 0.97, respectively. Even though the accuracy of ML methods was well-evaluated, their performance in tackling multi-cluster problems should be further examined.

Another widely used method to predict incoming floods is based on fuzzy models, such as TS and Mamdani. The capability of the TS model has been examined by Lohani et al. [15]. The TS model was compared to an ML technique called an artificial neural network (ANN) using rainfall rate and discharge data of a river in India. Their research, in particular, was designed to predict floods given higher-to-low flow region constraints. In this regard, the performance of the TS model was evaluated using a statistical model called peak percent threshold statistic. The results showed that the TS model outperformed the ANN.

In the research of Rostami et al. [16], data anomaly detection was added before processing the fuzzy inference system. Unlike the others that rely on sensory tools to infer incoming floods, their data anomaly detection consists of three algorithms: median-interquartile range, multi-layer perceptron, and recurrent neural network. Moreover, remote sensing climate data was used to generate early warnings of flood events. Two other researchers who showed the capability of the fuzzy-based approach to alert incoming floods are Jayawardena et al. [17] and Wee et al. [18]. Wee et al. [18] compared fuzzy with impact-based forecasting by developing a relation matrix between the level of risks and their impacts. However, trial-and-error in determining fuzzy sets and degree of membership must be conducted carefully to avoid performance degradation of the fuzzy inference system [17].

Even though previously mentioned fuzzy-based techniques to generate flood-related early warnings have already been well-proven, some issues still need to be solved, particularly in optimizing the construction of fuzzy rules. Lyu et al. [10] suggested combining fuzzy and machine-learning approaches to transform complex fuzzy inference rules or functions into a machine-learning model. However, when the fuzzy rules have been manually set, the impact of the machine-learning model to replace the fuzzy rules is relatively trivial as those rules are already reliable and can be used to infer the target state. The FLBL that can automatically learn and generate fuzzy rules becomes an option. This technique has been used in other domains, such as literature [22]. However, Bai and Lu [23] viewed that the fuzzy rules produced by FLBL should be optimized. In this regard, they proposed fuzzy set fusions. As it can increase complexities, a new strategy is needed to reduce fuzzy rules.

3. The Proposed River Flood EWS Using FLBL

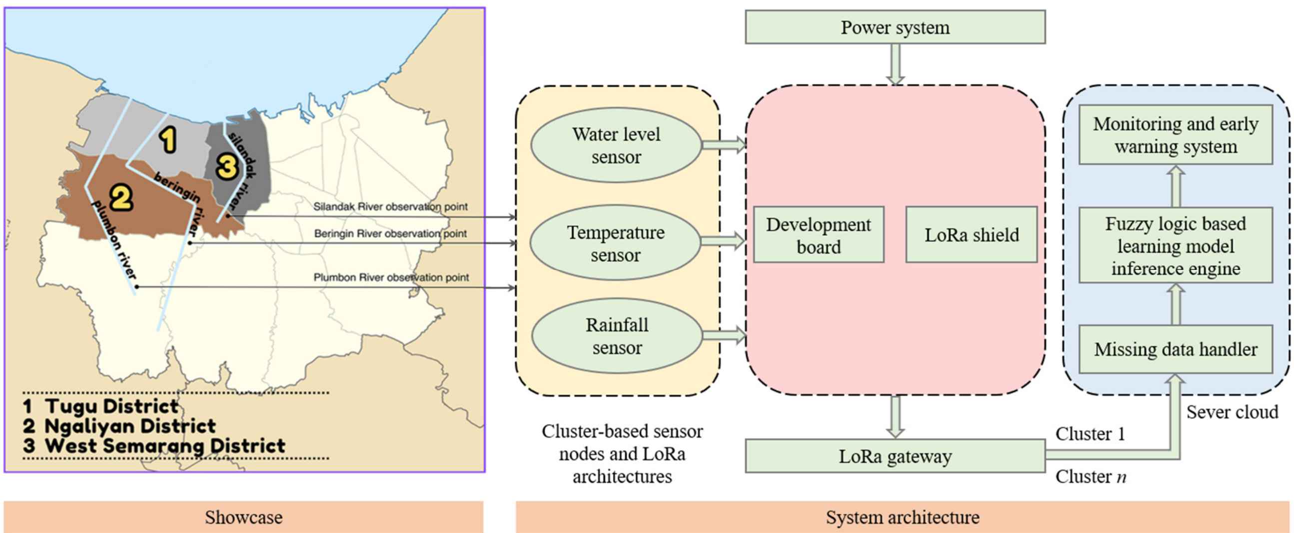


Fig. 1 The proposed system architecture for flood river early warning

This section explains the proposed river flood EWS exploiting FLBL. As illustrated in Fig. 1, there are two main blocks in the architecture of the proposed systems: the cluster-based IoT nodes and the server cloud. There are five processing phases in this proposal:

(1) The development board installed at a cluster collects the sensor reading values periodically based on the pre-defined settings.

(2) The collected data is sent to the cloud through the LoRa platform.

(3) At the server, a missing data handler will be activated if there is no data received in a determined time interval.

(4) The received data will be evaluated using the fuzzy rule model generated by the proposed FLBL.

(5) The flood prediction results will be forwarded to monitoring and EWS. The following subsections explain the details of the proposed system.

3.1. *Cluster-based sensor nodes*

This paper assumes that every observation point is a cluster, and every cluster is commonly associated with a specific observed region. However, an observation point may be close to another, particularly in river branches, and each branch has a potential flood impact on different areas. At an observation point, an IoT node is installed. This node comprises seven components, including a development board, water level sensor, temperature sensor, rainfall sensor, LoRa shield, LoRa gateway, and power system. The water level and rainfall rate sensors are the primary sources for calculating the chance of incoming river flood events in the proposed system. Each IoT node also has a temperature sensor, mainly used to confirm whether the rainfall rate sensors work well. Thus, this addition makes the proposed system more robust.

3.2. *Server-side processing*

This sub-section highlights two main parts of the server: the missing data handler and the FLBL inference engine. The output of FLBL is a fuzzy rule model. In generating the fuzzy rule model, FLBL has four steps, including the input data preparation, associating data to the corresponding fuzzy set fuzzification process, the learning process to generate fuzzy rules, and the rule optimization, as illustrated in Fig. 2. The first step processes the input data from sensors and changes the data into a fuzzy membership belonging to the corresponding fuzzy set. The linguistic term resulting from the fuzzification process was then mapped to produce several groups of rules. After that, the optimization process is performed to produce the final fuzzy rule model. This model infers incoming flood events in the EWS using real-time data. The detailed explanations are provided below.

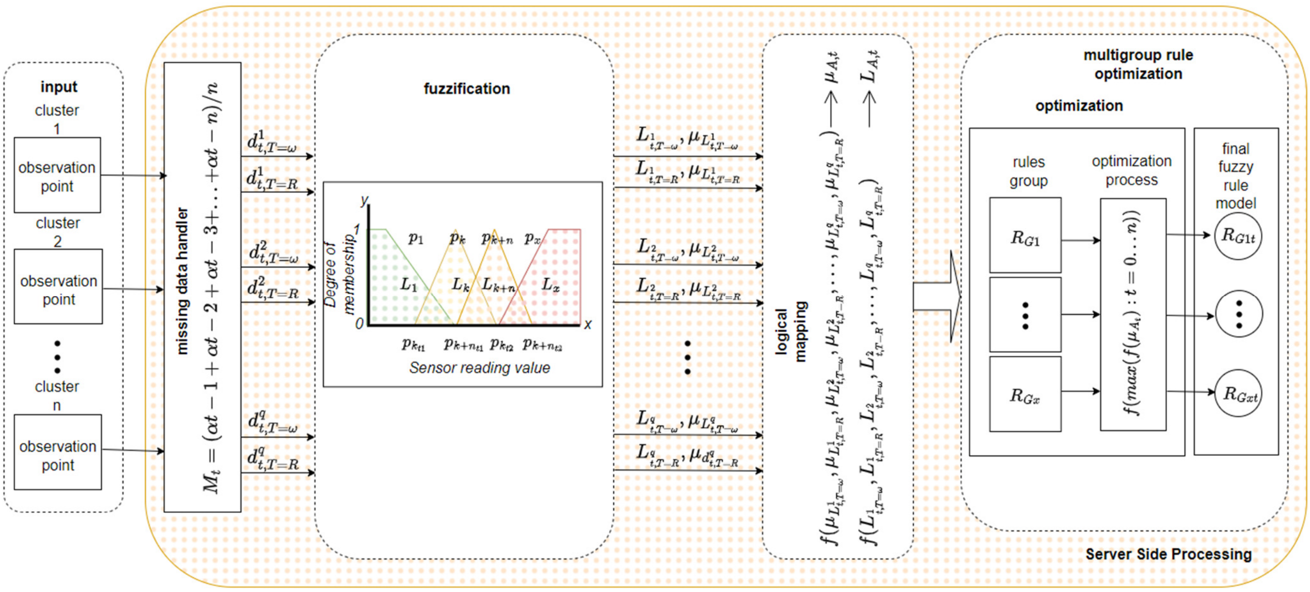


Fig. 2 Fuzzy logic-based learning

3.2.1. *Missing value handler*

The IoT-based architecture is vulnerable to data transmission failure. Therefore, the proposed approach implements a handler to impute sensor reading missing values. In this regard, the proposed system used a rolling statistical technique called simple moving average to impute missing values. Suppose that d is the data generated by a sensor at time t. The task of the simple moving average is aggregating previously captured values to replace the missing value at time $t(M_t)$ using the following:

$$M_t = \frac{\left(d_{t-1} + d_{t-2} + d_{t-3} + \ldots + d_{t-n}\right)}{n} \tag{1}$$

where n is the determined number of captured values from previous reading data.

3.2.2. *Generating fuzzy rules model*

A. Input data preparation:

As illustrated in Fig. 2, the learning process uses the data collected from sensors installed in the observation point cluster. Each sensor controller at the observation points periodically sends a tuple of data S_t^q, such that:

$$S_t^q = \left\{d_{t,T=\omega}^q, d_{t,T=R}^q\right\} \tag{2}$$

where d is the data generated by sensor T from the cluster number q at time t. There are two possible values of T based on the proposed EWS architecture: the water level sensor (denoted by ω) and the rainfall rate sensor (denoted by R). The temperature sensor is excluded from the dataset when generating the fuzzy rule model.

From the data collection, the universe of discourse U_{dT} is determined for each sensor type, such that:

$$U_{dT} = \left[\max\left(d_T\right), \min\left(d_T\right)\right] \tag{3}$$

where d_T is a set of data collected from a type of sensor at all clusters. Therefore, there will be $U_{dT=\omega}$ and $U_{dT=R}$. Furthermore, the dataset used for the further process (denoted by D) consists of the unification of S_t^q collected during a specific period such that:

$$D = \left\{\left(d_{1,T=\omega}^1, d_{1,T=R}^1\right), \left(d_{1,T=\omega}^2, d_{1,T=R}^2\right), \ldots, \left(d_{t+n,T=\omega}^q, d_{t+n,T=R}^q\right)\right\} \tag{4}$$

where n is the end period of collected data. The number of clusters and sensors in D is correlated with the number of attributes in the dataset. For example, if the number of clusters is 3 and there are 2 sensors in each cluster, then D will have six attributes.

B. Fuzzification process

In this step, $\forall d_j \in D$ will be fuzzified to associate them with the corresponding fuzzy set. Let $L = \{L_1, L_2, L_3, \cdots, L_x\}$ denotes a set of linguistic terms (e.g., low, normal, high, extreme) for fuzzy sets used to describe the reading value from a sensor, where x indicates the number of linguistic terms of the fuzzy sets. For each sensor, the proposed FLBL will determine the fuzzy sets based on two parameters, namely x and I denoting the desired interval value for each fuzzy set. Furthermore, the fuzzy sets indicated by the lowest and the highest x will have a trapezoid membership function. The intermediate fuzzy set(s) uses the triangular membership function.

Some critical points need to be determined by FLBL, following the number of linguistic terms used for each sensor (x). Those critical points (illustrated in Fig. 3) are:

(1) p_1 denoting the endpoint of the trapezoid membership function belongs to L_1 having $\mu_{L_1} = 1$

(2) p_x denoting the endpoint of the trapezoid membership function belongs to L_x having $\mu_{L_x} = 1$

(3) p_k denoting the top point of the triangular membership function belongs to L_k having $\mu_{L_k} = 1$, where k represents the infinite integer number of fuzzy sets having a triangular membership function located between L_1 and L_x, and $1 < k < x$

(4) $p_{k_{t1}}$ and $p_{k_{t2}}$ denoting the left and right threshold in the x-axis belongs to L_k, respectively, having $\mu_{L_k} = 0$

Now suppose that v_{max} and v_{min} are the minimum and maximum values in U_s, respectively, and $v \in s$. The points of p_1 and p_x are determined based on the value of v_{max}, v_{min}, and I as formulated below:

$$p_x = v_{max} - v_{max} \pmod I \tag{5}$$

$$p_1 = \frac{p_x}{x} \tag{6}$$

Furthermore, p_k is calculated as follows:

$$p_k = p_1 \times k \tag{7}$$

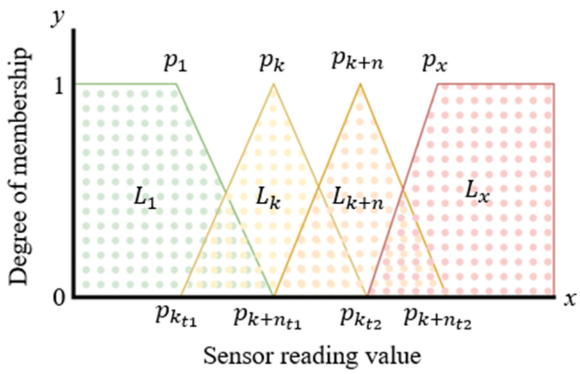


Fig. 3 Critical points of fuzzy sets

It should be noted that $min(k) = 2$ and $max(k) = x - 1$. Recalling that d_j represents the sensor reading value in the x-axis, the membership values of d_j in the fuzzy set having linguistic term L_x $\left(\mu_{d_j}^{Lx}\right)$ can be defined as follows:

$$\mu_{d_j}^{L_x} = \begin{cases} 1 & d_j \geq p_x \\ 0 & d_j \leq p_{\max(k)} \\ \left(d_j - p_{\max(k)}\right) / \left(p_x - p_{\max(k)}\right) & p_{\max(k)} < d_j < p_x \end{cases} \tag{8}$$

Similarly, the membership values of d_j in the fuzzy set having linguistic terms L_1 $\left(\mu_{d_j}^{L_1}\right)$ can be defined as follows:

$$\mu_{d_1}^{L_1} = \begin{cases} 1 & d_j \leq p_1 \\ 0 & d_j \geq p_{\min(k)} \\ \left(p_{\min(k)} - d_j\right) / \left(p_{\min(k)} - p_1\right) & p_1 < d_j < p_{\min(k)} \end{cases} \tag{9}$$

Next, $p_{k_{t1}}$ and $p_{k_{t2}}$ can be determined as follows:

$$p_{k_{t1}} = \begin{cases} p_1 & \min(k) \\ p_{k-1} & \min(k) < k < \max(k) \end{cases} \tag{10}$$

$$p_{k_{t2}} = \begin{cases} px & \max(k) \\ p_{k+1} & \min(k) \leq k < \max(k) \end{cases} \tag{11}$$

Thus, the membership values of d_j in the fuzzy set having linguistic terms L_k $\left(\mu_{d_j}^{L_k}\right)$ can be formulated as follows:

$$\mu_{d_j}^{L_k} = \begin{cases} 1 & d_j = p_k \\ 0 & d_j \le p_{k_{t1}} \ or \ d_j \ge p_{k_{t2}} \\ \left(d_j - p_{k_{t1}}\right) \Big/ \left(p_k - p_{k_{t1}}\right) & p_{k_{t1}} < d_j < p_k \\ \left(p_{k_{t2}} - d_j\right) \Big/ \left(p_{k_{t2}} - p_k\right) & p_k < d_j < p_{k_{t2}} \end{cases} \tag{12}$$

At the end of this stage, it can be assumed that at time t, $\forall d_{t,T=\omega}^{q} \in D$ and $\forall d_{t,T=R}^{q} \in D$, a linguistic term has been set, namely $L_{t,T=\omega}^{q}$ and $L_{t,T=R}^{q}$, respectively, including their degree of membership on the corresponding fuzzy set $\mu_{L_{t,T=\omega}^{q}}$ and $\mu_{L_{t,T=R}^{q}}$, respectively.

For further processing, the dataset$D$, which was previously defined in Eq. (4), will be replaced with the fuzzified dataset(D'). It should be noted that besides representing data collection time, t can also be viewed as the row number in the dataset. Now, it can be assumed that D_t' is one row of the fuzzified dataset, and it contains two tuples of information from each cluster at time t, such that

$$D_t' = \left\{\left[\left(L_{1,T=\omega}^{1}, \mu_{1,T=\omega}^{1}\right), \left(L_{1,T=R}^{1}, \mu_{1,T=R}^{1}\right)\right], \ldots, \left[\left(L_{t+n,T=\omega}^{q}, \mu_{t+n,T=\omega}^{q}\right), \left(L_{t+n,T=R}^{q}, \mu_{t+n,T=R}^{q}\right)\right]\right\} \tag{13}$$

The information in D' will be used to feed the FLBL learning process.

C. Fuzzy logic-based learning process

The FLBL learning process is initiated after the input dataset is fuzzified into its linguistic term. The target of the learning process is to optimize fuzzy rules from the given dataset. There are two steps in the learning process: (1) producing rules using a logical mapping function and (2) performing fuzzy rules optimization. While producing rules, there are two terms: the antecedent and the consequent, denoted by ∂ and τ, respectively. The former is the assigned degree of membership from input sensor values. In other words, ∂ at time $t(\partial_t)$ is constructed as follows:

$$\partial_t = \left\{\mu_{L_{1,T=\omega}^{1}}, \mu_{L_{1,T=R}^{1}}, \mu_{L_{1,T=\omega}^{2}}, \mu_{L_{1,T=R}^{2}}, \ldots, \mu_{L_{t+n,T=\omega}^{q}}, \mu_{L_{t+n,T=R}^{q}}\right\} \tag{14}$$

Furthermore, τ_t is the output variable that can be the assigned label (A_t) for each row in the dataset D'. The fuzzification process is also applied for A_t resulting in its linguistic term denoted by L_{A_t} and its degree of membership in the corresponding fuzzy set (μ_{A_t}). The fuzzy rule generation is structured based on the relation of ∂_t and τ_t, and is determined as follows:

$$f\left\{\mu_{L_{1,T=\omega}^{1}}, \mu_{L_{1,T=R}^{1}}, \mu_{L_{1,T=\omega}^{2}}, \mu_{L_{1,T=R}^{2}}, \ldots, \mu_{L_{t+n,T=\omega}^{q}}, \mu_{L_{t+n,T=R}^{q}}\right\} \rightarrow \mu_{A_t} \tag{15}$$

Based on the logical mapping function above, the linguistic terms of the antecedent can be mapped to the consequent for each row in the dataset, such that:

$$f\left\{L_{1,T=\omega}^{1}, L_{1,T=R}^{1}, L_{1,T=\omega}^{2}, L_{1,T=R}^{2}, \ldots, L_{t+n,T=\omega}^{q}, L_{t+n,T=R}^{q}\right\} \rightarrow L_{A_t} \tag{16}$$

Now, the fuzzy rules for the specific time t can be generated by forming and functions among the antecedent's linguistic terms with L_{A_t} as the returning value.

The second process in the FLBL learning is performing fuzzy rules optimization. This process is conducted through several steps. The first step is the unification of three sets μ_A, L_A, and D'. The sets μ_A and L_A are sets of consequent linguistic terms and their degree of membership generated by Eq. (15) and Eq. (16), respectively. The matching process uses index matching driven by t. Before this step is executed, the pre-requisite must be met which can be defined as follows:

$$|L_A| = |D^{'}| = |\mu_A| \tag{17}$$

Which means that the size of both sets must be equal. Once the pre-requisite is fulfilled, the unification step is initiated to produce a set of fuzzy rules (FR), such that:

$$FR_t = \{D_t^{''}, L_{A_t}\} \tag{18}$$

where t indicates the row number. Meanwhile, $D_t^{''}$is similar to D_t' with the degree of membership excluded from the dataset, remaining only the label ($L_{t,T=\omega}^q$ and $L_{t,T=R}^q$). The size of FR(|FR|) is considered the total of initial fuzzy rules that will be optimized by reducing them. The optimization strategy contains two main procedures: rules grouping and rules fusion. The grouping procedure follows a similarity check called the SSC which can be determined as follows:

$$\text{SSC} = \frac{|R_G \cap \text{FR}_t|}{\min(|R_G|, |\text{FR}_t|)} \tag{19}$$

Firstly, the grouping procedure will insert the initial row ($\text{FR}_{t=0}$) to a rule group set (R_G) in which G indicates the group number. Hence, $R_{G=0}$ will have the same rule value compared to $\text{FR}_{t=0}$. In the second loop, the grouping procedure will compare the FR value for the next t (e.g., $t = 1$) to each existing R_G. When the SSC value is 1, no R_G will be formed, and the rule will be represented, e.g., $\text{FR}_{t=1}$ will be added as a new element to the corresponding R_G. Otherwise, new R_G (e.g., $R_{G=1}$) will be created. Thus, the total rule group can be determined by $G + 1$.

After all, rows have been examined for the grouping process, it can be assumed that each R_G contains a set of the same rules, but they have different membership degrees in the consequent part. In the proposed approach, the rule having the highest membership degree will be selected from each R_G. The rule optimization is regulated by the following operation:

$$R_{G_t} = f\left(\max\left(f\left(\mu_{A_t}\right) : t = 0 \ldots n\right)\right) \tag{20}$$

where t is the row index, and R_{G_t} is the selected rule in a certain R_G having index t and the highest membership degree. Thus, the optimized fuzzy rule model (M) can be described as follows:

$$M = \{R_{G=0}, R_{G=1}, \ldots, R_{G=k_t}\} \tag{21}$$

where k is the rule group number.

4. Experiment and Result

This section presents an experiment and the result of this research and is divided into six sub-sections. The first sub-section explains the showcase used in this research, which uses three rivers in Semarang City, e.g., Bringin, Plumbon, and Silandak rivers. The second section describes the dataset collected from the Pemali Juana Office, the government's technical operation unit managing the Central Java River in Indonesia during 2020-2021. The next subsection demonstrates the implementation and performance of the FLBL system in this research. The final sub-sections consist of a discussion of this system and the future work that can be developed further from this research.

4.1. The showcase area

Table 1 displays the characteristics of three rivers in Semarang. Silandak is the shortest river, while the other two are similar in length. There is a significant difference in the water debit of the three rivers, with Bringing having the highest at 381.74 m^3/sec, followed by Plumbon with 244 m^3/sec, and the lowest being Silandak with 130.32 m^3/sec.

Table 1 River profile in the Tugu area

River name	Length (km)	Water debit (m^3/sec)
Plumbon river	19.5	244
Bringin river	21.6	381.74
Silandak river	10.88	130.32

Meanwhile, Table 2 presents the observed water travel times for the three rivers and regions with flooding risk. Each river has three observation points, each with a corresponding affected area. All three observation points potentially affected the Ngaliyan district. Two observation points, Plumbon and Silandak, have the same travel time to the affected area in 30 minutes, and Bringin has the longest travel time of 120 minutes. When a higher risk water level is detected, river flooding could potentially impact certain regions, particularly the districts near the observation points. For example, if the water level at the Plumbon Dam observation point rises to a high level, it could adversely affect the Ngaliyan district.

Table 2 Flood-affected area

Observation point	Affected area	Traveling time
Plumbon	Ngaliyan district	30 minutes
Bringin	Ngaliyan and Tugu district	120 minutes
Silandak	Ngaliyan and West Semarang district	30 minutes

4.2. The dataset

The dataset used to train the fuzzy rules model was collected from the Pemali Juana Office, the government's technical operation unit managing the Central Java River in Indonesia during 2020-2021. The dataset has three main attributes for each observation point: water level, rainfall rates, and temperature. However, the fuzzy rules generation only considers water level and rainfall rate data. The visualization of rainfall rates and water level data in the dataset can be seen in Fig. 4 and Fig. 5, respectively.

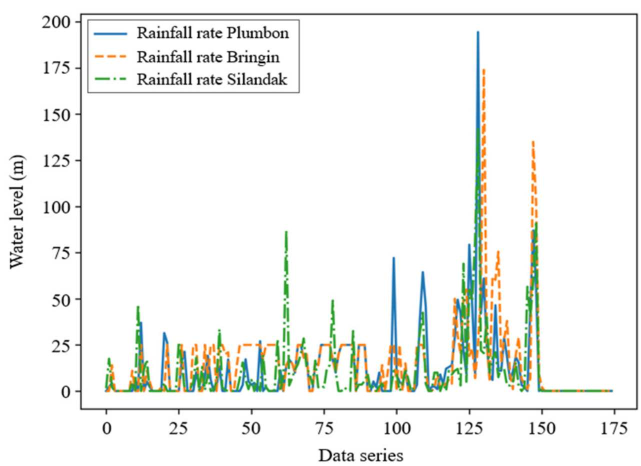


Fig. 4 Rainfall rates at three observation points: Plumbon, Bringin, and Silandak Dams in 2020-2021

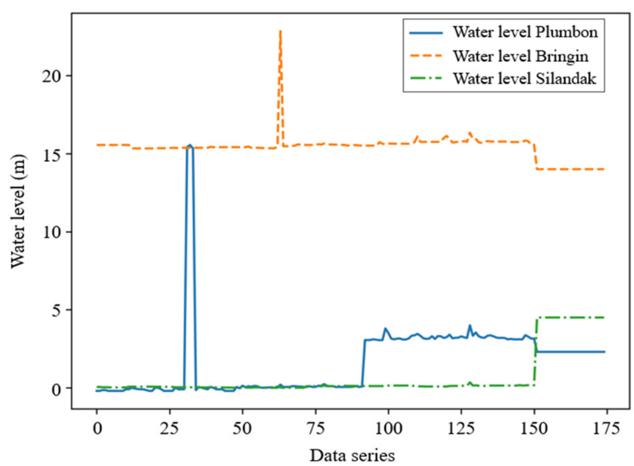


Fig. 5 Water level at three observation points: Plumbon, Bringin, and Silandak Dams in 2020-2021

4.3. The data imputation performance

This section illustrates how the missing data from sensors were handled in the proposed system as written in Eq. (1). In this research's dataset, approximately 5% of missing data from the sensors were found. The missing data were mostly due to connectivity problems. Table 3 shows a replaced value assuming that at $t = 0$, the data from sensors were missing. By examining the data from $t - 5$ to $t - 1$, the predicted value from the water level sensor at $t = 0$ was 0.118. Meanwhile, the predicted value for the rainfall rate sensor was 6.62. Those values are used to replace the missing data.

Table 3 Imputing missing data

Sensors	$t-5$	$t-4$	$t-3$	$t-2$	$t-1$	$t=0$
Water level	0.1	0.07	0.12	0.1	0.2	0.118*
Rainfall rate	3.3	17.2	5.1	3.4	4.1	6.62*

Note: * = a new value replacing missing data

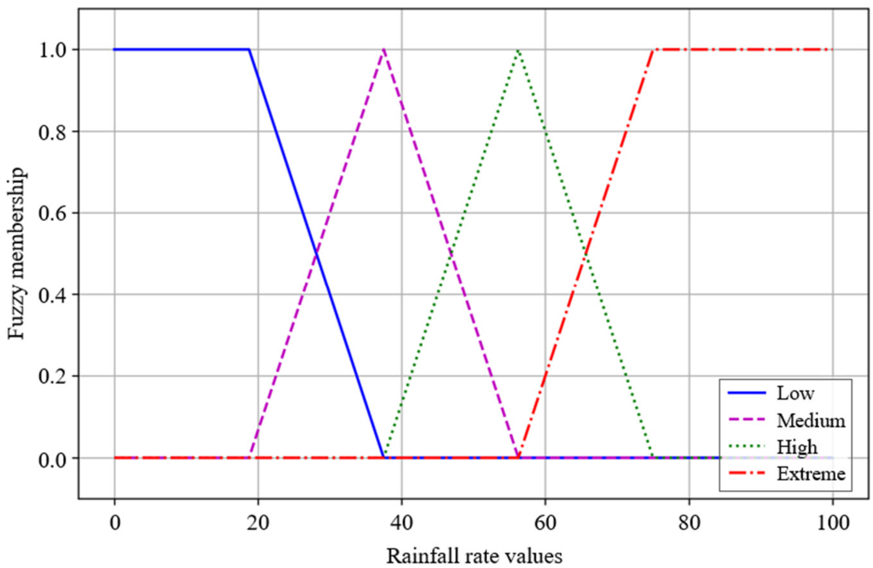
4.4. Generating fuzzy membership and fuzzy rules

Before generating fuzzy membership, both fuzzy sets and their linguistic terms were defined for the main attributes (water level and rainfall rate) used to predict flood events. Additionally, this research set a fuzzy set for the incoming flood-event alert level. As described in Table 4, rainfall rates and the water level had similar linguistic terms for their fuzzy sets: low, medium, and high. However, the rainfall rates had another fuzzy set called extreme. Furthermore, the alert levels have three linguistic terms for their fuzzy set: alert, normal, and danger. The alert level is the flood-event label assigned for combinations of the main attributes based on historical data.

Table 4 List of linguistic terms for each sensor

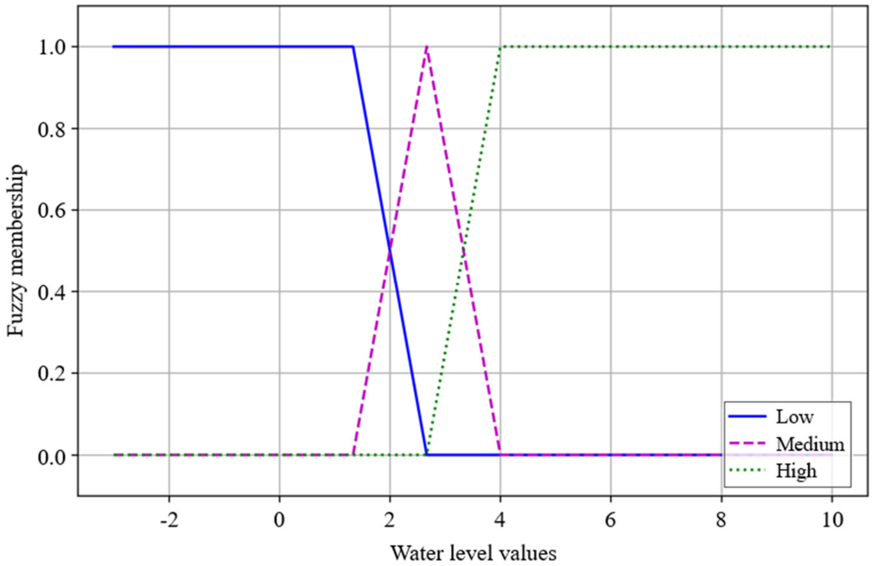
Attributes	Linguistic terms and symbols
Rainfall rates	Low (L) — Medium (M) — High (H) — Extreme (E)
Water level	Low (L) — Medium (M) — High (H)
Alert level	Normal (N) — Alert (A) — Danger (D)

Next was the learning process to generate fuzzy rules. It should be noted that from the three attributes in Table 4, the rainfall rates and water level are the antecedent, and the alert level is the consequent. For the starter, the proposed approach calculated the universe of discourse of each attribute by using Eq. (3). After that, critical points were calculated using Eqs. (5)-(7) based on the number of fuzzy sets determined for each attribute in the dataset. As the rainfall rate had four fuzzy sets (see Table 4), there were four critical points as references for two trapezoidal and two triangular membership functions (see Fig. 6(a). Differently, the water and alert levels had only three fuzzy sets, so only three critical points were set for two trapezoidal membership functions and one triangular membership function as shown in Fig. 6(b) and Fig. 6(c).

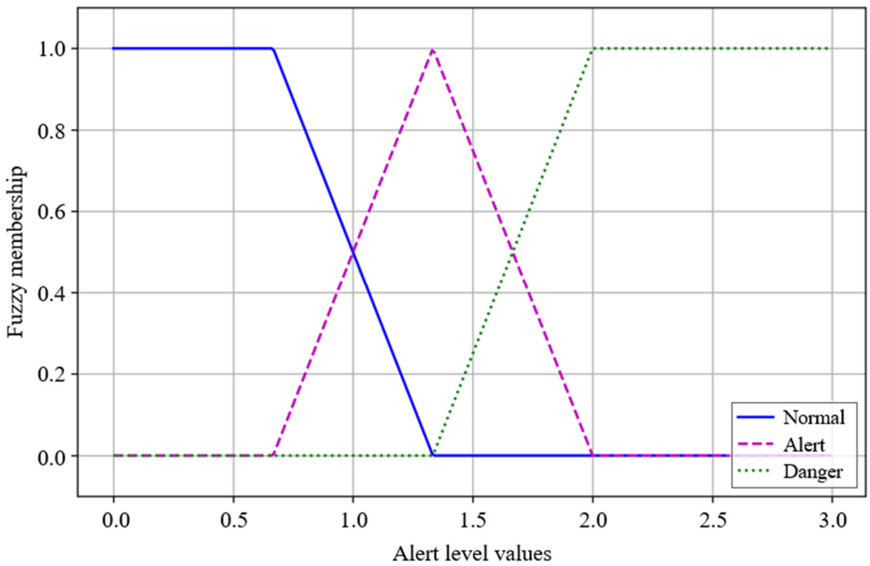


(a) Fuzzy membership of rainfall rate values

Fig. 6 The fuzzy membership values for main attributes



(b) Fuzzy membership of water level values



(c) Fuzzy membership of alert levels

Fig. 6 The fuzzy membership values for main attributes (continued)

Once the critical points were set for each attribute, their linguistic terms can be determined, including their degree of membership by using Eqs. (8)-(13). For example, Table 5 shows an example of rainfall rate data mapped to their corresponding fuzzy set's linguistic terms. Similarly, Table 6 provides the linguistic terms for water level data and its degree of membership.

Table 5 Example data mapping rainfall rate values to their corresponding fuzzy sets

Sensors values	Low	Medium	High	Extreme	Linguistic terms	Membership degree
2.3	1	0	0	0	Low	1
4.3	1	0	0	0	Low	1
19.1	0.98	0.02	0	0	Low	0.98
37	0.03	0.97	0	0	Medium	0.97
31.4	0.33	0.67	0	0	Medium	0.67
46	0	0.55	0.45	0	Medium	0.55
56.2	0	0.01	0.99	0	High	0.99
49.5	0	0.36	0.64	0	High	0.64
64.4	0	0	0.57	0.43	High	0.57
87	0	0	0	1	Extreme	1
72.1	0	0	0.15	0.85	Extreme	0.85
79.3	0	0	0	1	Extreme	1

Table 6 Example data mapping water level values to their corresponding fuzzy sets

Sensor values	Low	Medium	High	Linguistic term	Membership degree
-0.2	1	0	0	Low	1
0.23	1	0	0	Low	1
0.1	1	0	0	Low	1
2.3	0.28	0.72	0	Medium	0.72
3.17	0	0.62	0.38	Medium	0.62
3.24	0	0.57	0.43	Medium	0.57
3.36	0	0.48	0.52	High	0.52
3.45	0	0.41	0.59	High	0.59
4	0	0	1	High	1

Regarding the alert levels, as previously mentioned in Eqs. (15)-(16), they were mapped into three linguistic terms, and their degree membership can be seen in Table 7. The alert level 0 has a full (1) degree membership value in the ‘Normal’ fuzzy set. Furthermore, alert levels 1 and 2 had half (0.5) and total (1) degree membership values in the ‘Alert’ and ‘Danger’ fuzzy sets, respectively.

Table 7 Mapping alert level values to their corresponding fuzzy sets

Alert level	Normal	Alert	Danger	Linguistic term	Membership degree
0	1	0	0	Normal	1
1	0.5	0.5	0	Alert	0.5
2	0	0	1	Danger	1

After all the sensor values were mapped into linguistic terms related to corresponding fuzzy sets, the proposed method summarized the fuzzy rules extracted from the relation between the antecedent and the consequent. The example of fuzzy rules generated by the proposed method can be seen in Table 8. The fuzzy rules were region-based, meaning that West Semarang, Tugu, and Ngaliyan districts might have different alert levels even though they had the same antecedent.

Table 8 Example of generated fuzzy rules for flood-event alert in west Semarang district

RP	RB	RS	WP	WB	WS	Alert level
E	E	H	H	H	L	D
E	L	H	M	H	L	A
H	E	E	M	H	L	D
M	L	M	M	H	L	N
M	L	H	M	H	L	A
L	L	H	L	H	L	A

Notes:
(1) RP, RB, RS: Rainfall rate for Plumbon, Bringin, and Silandak, respectively
(2) WP, WB, WS: Water level for Plumbon, Bringin, and Silandak, respectively
(3) Other symbols see Table 4

4.5. Results and discussion

This section provides performance comparison results between the baseline method from previous studies and the proposed method. Two baseline methods by Vinothini and Jayanthy [4] and Anbarasan et al. [13] were selected, and they used the decision tree and neural network algorithms, respectively, to trigger the EWS alert. The decision tree method was selected because it generates a set of rules, and the performance was comparable to the fuzzy rules from the proposed method. Meanwhile, the neural network approach was selected as it is well-proven in a previous study.

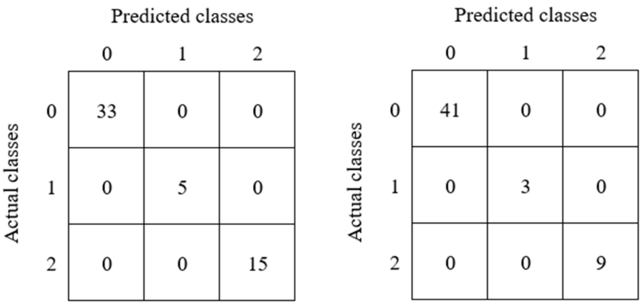
The results can be seen in Table 9, which shows five trials conducted using the proposed method and two baseline methods, as mentioned. From the results, the proposed method in most districts in all five trials achieved the highest accuracy except in the second trial in the Tugu district and the fourth trial in the Ngaliyan district. The accuracy of the proposed method was 96% in those trials, while the baseline method using the decision tree showed 98% accuracy in comparison. However, the other trials in three districts showed that the proposed method was more accurate.

Table 9 Prediction performance

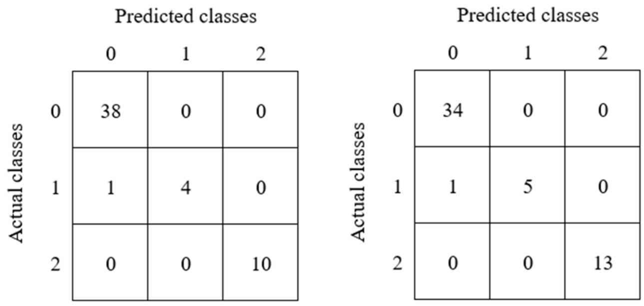
Trial no.	West Semarang			Tugu			Ngaliyan		
	A	B	C	A	B	C	A	B	C
1	0.98	0.98	0.92	0.98	0.96	0.64	0.98	0.92	0.64
2	1	0.98	0.94	0.96	0.98	0.64	0.94	0.94	0.64
3	1	0.98	0.88	1	0.96	0.61	0.98	0.96	0.64
4	1	0.98	0.97	0.96	0.92	0.63	0.96	0.98	0.64
5	1	0.96	0.91	0.96	0.94	0.62	0.98	0.96	0.64
Score	0.996	0.976	0.924	0.972	0.952	0.628	0.968	0.952	0.64
#Rules	29	4	N/A	29	4	N/A	29	8	N/A

Notes:
A = Proposed method; B = Baseline method [4];
C = Baseline method [13]; N/A = Not available

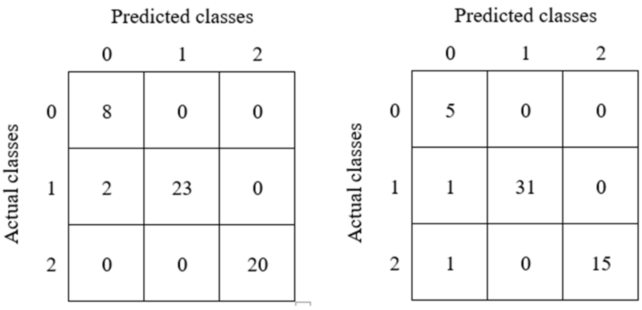
In the districts of Tugu, Ngaliyan, and Western Semarang, the suggested method's average accuracy ratings were 99.6%, 97.2%, and 96.8%, respectively. The decision tree achieved an average accuracy of 97.6%, 95.2%, and 95.2%, respectively, compared to the baseline approaches. The average accuracy of the neural network approach was 92.4%, 62.8%, and 64% for each of the three selected districts. The proposed approach demonstrated the highest average accuracy. Furthermore, Fig. 7 compares the confusion matrix between the proposed FLBL method and the decision tree method from Vinothini and Jayanthy [4] based on trials 2 and 4.



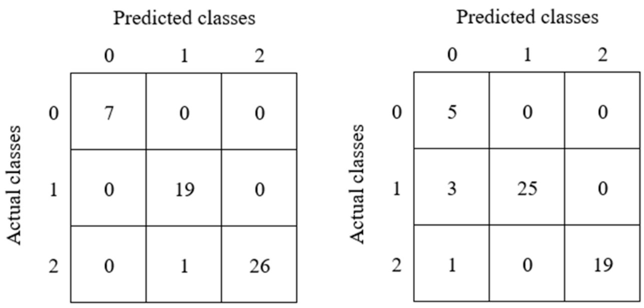
(a) FLBL confusion matrix trial number 2 and 4 in predicting Western Semarang district



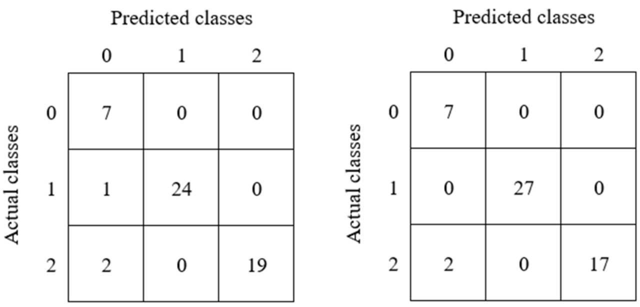
(b) Decision tree confusion matrix trial number 2 and 4 in predicting Western Semarang district



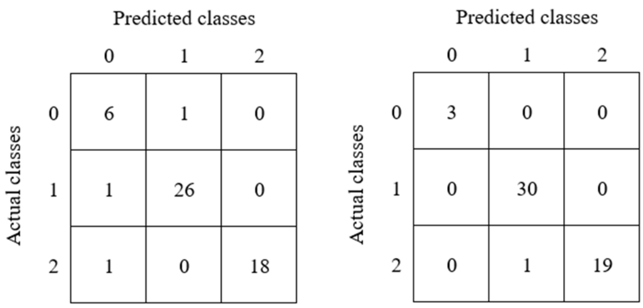
(c) FLBL confusion matrix trial number 2 and 4 in predicting in Predicting Tugu district



(d) Decision tree confusion matrix trial number 2 and 4 in predicting Tugu district



(e) FLBL confusion matrix trial number 2 and 4 in predicting Ngaliyan district



(f) Decision tree confusion matrix trial number 2 and 4 in predicting Ngaliyan district

Fig. 7 Confusion matrix comparison based on trial no. 2 and trial no. 4

In the West Semarang District, FLBL generated no false predictions (see Fig. 7(a)), while the compared method indicated one mistake in which the 'alert' class was mistakenly labeled as 'normal' (see Fig. 7(b)). In the Tugu district, FLBL has mistakenly identified some 'alert' classes with 'normal' and 'danger' with 'normal' (see Fig. 7(c)). The decision tree method had similar mistakes in this district, but with more false label predictions such as 'alert' to 'normal'; 'danger' to 'normal'; and 'danger' to 'alert' (see Fig. 7(d)). While in the previous two districts, the proposed FLBL method outperformed the decision tree, the contrary occurred in the Ngaliyan district. In total, FLBL generated 5 mistakes (see Fig. 7(e)), higher than the compared method that only had 3 mistakes (see Fig. 7(f)). However, overall, in all trials, the proposed method had lower mistakes between the actual and the predicted classes compared to the decision tree.

Based on these experiments, it can be concluded that the proposed method outperformed the baseline methods in terms of accuracy, so it can reduce false alarms to notify flood events in the EWS. Regarding the rule optimization process, the proposed method had 29 final fuzzy rules to achieve in all districts, while the baseline method using a decision tree had 4 final rules for West Semarang district and Tugu district but had 8 final rules for Ngaliyan district. The rules were not available in the baseline method using a Neural Network.

Comparing the self-determined fuzzy rules approach with the proposed FLBL method, the former resulted in 1728 fuzzy rules, while the latter only produced 29. Considering this rule gap, the proposed approach has benefits and drawbacks. When most of the incoming input data has already been characterized in the training dataset, the proposed method can optimize fuzzy rules. In the meantime, like the other ML approaches, the performance of the proposed approach will be poor when the training dataset does not reflect the daily incoming data.

4.6. Future works

This study offers the development of a learning-based fuzzy rules model to enhance the prediction of incoming floods. However, this research highlighted some areas for further improvement, particularly to resolve two main weaknesses of the proposal. The first weakness is due to the reliability of the IoT nodes. For an EWS system, all IoT devices must be prepared for a sudden disaster by being on standby for 24 hours. The IoT node is supported by a power system consisting of a solar panel, battery, and solar charge controller, and it can survive for several days. However, the power system cannot support the battery when the weather is cloudy. If this condition persists for days, it will become a problem for the IoT hardware. With lots of connected sensors, no power means that there is a possibility that the development board will have missing or duplicate data due to the radio frequency connection loss and the unavailability of the LoRa gateway. Minimizing such problems could be considered a potential future work by implementing several strategies, including circular buffers and sliding window protocol.

Furthermore, missing data problems can also occur from the LoRa gateway to the server side, which can be handled by setting its quality of service (QoS) to level 2 using message queuing telemetry transport (MQTT) protocol to obtain data later when the devices are back online. In this research, missing data problems during data transmission to the server side have been tackled. However, they may cause a critical impact when they occur in sequences, and the number of sequences is bigger than the threshold values representing several captured values from previous reading data as shown in Eq. (1).

Furthermore, the next weakness comes from the FLBL method. The normal water level of a particular river is classified as 'high' compared to the other rivers (see Fig. 5, Bringin water level). The automation of the fuzzification process needs to be improved to address such problems. Besides overcoming the weaknesses, future studies could consider the potential key features, particularly by taking advantage of other data sources, such as Synthetic Aperture data from radars. Even though it may increase the EWS cost, it can provide good support for developing runoff simulation. Another future improvement could be directed to enhance early warning propagation strategies utilizing various platforms, such as mobile devices that allow access to real-time information and flood alerts.

5. Conclusion

This paper presents an IoT-based river flood EWS solution, including the hardware architecture, communication protocol, and a novel FLBL method to predict incoming floods. The proposed FLBL method involves two other techniques: a simple moving average to impute missing data and a join function between SSC similarity and max functions as the fuzzy rules optimization strategy. The proposed FLBL generates a fuzzy rules model from a given dataset. The solution was evaluated at three observation points: Plumbon, Silandak, and Bringin river, traversing three districts in Semarang City: Ngaliyan, Tugu, and West Semarang.

The results showed that 5% of missing data from the dataset can be handled by the moving average method without causing a significant impact on the FLBL model's accuracy. The results from the showcase indicated that the proposed method was more accurate in predicting incoming floods compared to the baseline methods used in this study, which had an average of 97.8% accurate predictions. Meanwhile, the decision tree and the Neural Network delivered 96% and 73.07% accurate predictions, respectively. Furthermore, although the FLBL method generates more rules (29 rules) than the decision tree (4-8 rules), its accuracy is not significantly affected. However, the proposed FLBL still had some weaknesses. The method was designed to have two trapezoidal fuzzy membership functions with triangular membership functions in between. With such a design, it could not have other types of membership functions. Future direction can be directed to enhance the fuzzy set creation process.

Acknowledgment

This research is funded by the Faculty of Engineering, Diponegoro University, through Strategic Research Program No. 119/UN7.F3/HK/V/2023.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] R. B. Mudashiru, N. Sabtu, I. Abustan, and W. Balogun, "Flood Hazard Mapping Methods: A Review," Journal of Hydrology, vol. 603, part A, article no. 126846, December 2021.

[2] A. Lamers, S. Devi. S, M. Sharma, R. Berg, J. M. Gálvez, Z. Yu, et al., "Forecasting Tropical Cyclone Rainfall and Flooding Hazards and Impacts," Tropical Cyclone Research and Review, vol. 12, no. 2, pp. 100-112, June 2023.

[3] M. S. S. Sayyad, P. Surve, N. Shaikh, M. Gharat, and P. Tambe, "IoT Based Early Flood Detection and Avoidance," Iconic Research and Engineering Journals, vol. 3, no. 12, pp. 50–55, June 2020.

[4] K. Vinothini and S. Jayanthy, "IoT Based Flood Detection and Notification System Using Decision Tree Algorithm," International Conference on Intelligent Computing and Control Systems, pp. 1481-1486, May 2019.

[5] M. I. Zakaria, W. A. Jabbar, and N. Sulaiman, "Development of a Smart Sensing Unit for LoRaWAN-Based IoT Flood Monitoring and Warning System in Catchment Areas," Internet of Things and Cyber-Physical Systems, vol. 3, pp. 249-261, 2023.

[6] A. Radhika, G. Manochitra, K. Keerthanen, K. Mahendran, R. M. Kumar, and N. M. Sarath, "Low Cost IoT Based Early Detection of Flood Monitoring and Alert System Using GPS and GSM," AIP Conference Proceedings, vol. 2527, no. 1, October 2022.

[7] K. Malek, E. Ortiz-Rodriguez, Y. C. Lee, J. Murillo, A. Mohammadkhorasani, L. Vigil, et al., "Design and Implementation of Sustainable Solar Energy Harvesting for Low-Cost Remote Sensors Equipped with Real-Time Monitoring Systems," Journal of Infrastructure Intelligence and Resilience, vol. 2, no. 3, article no. 100051, September 2023.

[8] A. I. Silverman, T. Brain, B. Branco, P. sai venkat Challagonda, P. Choi, R. Fischman, et al., "Making Waves: Uses of Real-Time, Hyperlocal Flood Sensor Data for Emergency Management, Resiliency Planning, and Flood Impact Mitigation," Water Research, vol. 220, article no. 118648, July 2022.

[9] Y. Liu, X. K. Wang, W. H. Hou, H. Liu, and J. Q. Wang, "A Novel Hybrid Model Combining a Fuzzy Inference System and a Deep Learning Method for Short-Term Traffic Flow Prediction," Knowledge-Based Systems, vol. 255, article no. 109760, November 2022.

[10] H. M. Lyu, Z. Y. Yin, A. Zhou, and S. L. Shen, "MCDM-Based Flood Risk Assessment of Metro Systems in Smart City Development: A Review," Environmental Impact Assessment Review, vol. 101, article no. 107154, July 2023.

[11] R. Costache, M. C. Popa, D. T. Bui, D. C. Diaconu, N. Ciubotaru, G. Minea, et al., "Spatial Predicting of Flood Potential Areas Using Novel Hybridizations of Fuzzy Decision-Making, Bivariate Statistics, and Machine Learning," Journal of Hydrology, vol. 585, article no. 124808, June 2020.

[12] E. Šarak, M. Dobrojević, and S. A. Sedmak, "IoT Based Early Warning System for Torrential Floods," FME Transactions, vol. 48, no. 3, pp. 511-515, 2020.

[13] M. Anbarasan, B. Muthu, C. B. Sivaparthipan, R. Sundarasekar, S. Kadry, S. Krishnamoorthy, et al., "Detection of Flood Disaster System Based on IoT, Big Data and Convolutional Deep Neural Network," Computer Communications, vol. 150, pp. 150-157, January 2020.

[14] S. I. Abdullahi, M. H. Habaebi, and N. Abd Malik, "Intelligent Flood Disaster Warning on the Fly: Developing IoT-Based Management Platform and Using 2-Class Neural Network to Predict Flood Status," Bulletin of Electrical Engineering and Informatics, vol. 8, no. 2, pp. 706-717, June 2019.

[15] A. K. Lohani, N. K. Goel, and K. K. S. Bhatia, "Improving Real Time Flood Forecasting Using Fuzzy Inference System," Journal of Hydrology, vol. 509, pp. 25-41, February 2014.

[16] A. Rostami, M. Akhoondzadeh, and M. Amani, "A Fuzzy-Based Flood Warning System Using 19-Year Remote Sensing Time Series Data in the Google Earth Engine Cloud Platform," Advances in Space Research, vol. 70, no. 5, pp. 1406-1428, September 2022.

[17] A. W. Jayawardena, E. D. P. Perera, B. Zhu, J. D. Amarasekara, and V. Vereivalu, "A Comparative Study of Fuzzy Logic Systems Approach for River Discharge Prediction," Journal of Hydrology, vol. 514, pp. 85-101, June 2014.

[18] G. Wee, L. C. Chang, F. J. Chang, and M. Z. M. Amin, "A Flood Impact-Based Forecasting System by Fuzzy Inference Techniques," Journal of Hydrology, vol. 625, part. B, article no. 130117, October 2023.

[19] S. Supatmi, R. Hou, and I. D. Sumitra, "Study of Hybrid Neurofuzzy Inference System for Forecasting Flood Event Vulnerability in Indonesia," Computational Intelligence and Neuroscience, vol. 2019, no. 1, article no. 6203510, 2019.

[20] B. Agbo, H. Al-Aqrabi, R. Hill, and T. Alsboui, "Missing Data Imputation in the Internet of Things Sensor Networks," Future Internet, vol. 14, no. 5, article no. 143, May 2022.

[21] Q. Shao, A. Aldhafeeri, S. Qiu, and S. Khuder, "A Multiplicative Holt–Winters Model and Autoregressive Moving-Average for Hyponatremia Mortality Rates," Healthcare Analytics, vol. 4, article no. 100262, December 2023.

[22] C. K. Leung, J. D. Elias, S. M. Minuk, A. R. R. de Jesus, and A. Cuzzocrea, "An Innovative Fuzzy Logic-Based Machine Learning Algorithm for Supporting Predictive Analytics on Big Transportation Data," IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-8, July 2020.

[23] Y. Bai and X. Lu, "Multiple Kernel Learning-Based Rule Reduction Method for Fuzzy Modeling," Fuzzy Sets and Systems, vol. 465, article no. 108534, August 2023.

[24] J. Ringo, S. Sabai, and A. Mahenge, "Performance of Early Warning Systems in Mitigating Flood Effects. A Review," Journal of African Earth Sciences, vol. 210, article no. 105134, February 2024.

[25] M. Siddique, T. Ahmed, and M. S. Husain, "Flood Monitoring and Early Warning Systems – An IoT Based Perspective," EAI Endorsed Transactions on Internet of Things, vol. 9, no. 2, article no. e4, 2023.

[26] H. Z. Tedla, T. W. Bekele, L. Nigussie, E. D. Negash, C. L. Walsh, G. O'Donnell, et al., "Threshold-Based Flood Early Warning in an Urbanizing Catchment Through Multi-Source Data Integration: Satellite and Citizen Science Contribution," Journal of Hydrology, vol. 635, article no. 131076, May 2024.

[27] E. Gambini, A. Ceppi, G. Ravazzani, M. Mancini, I. Q. Valsecchi, A. Cucchi, et al., "An Empirical Rainfall Threshold Approach for the Civil Protection Flood Warning System on the Milan Urban Area," Journal of Hydrology, vol. 628, article no. 130513, January 2024.

[28] C. Prakash, A. Barthwal, and D. Acharya, "FLOODALERT: An Internet of Things Based Real-Time Flash Flood Tracking and Prediction System," Multimedia Tools and Applications, vol. 82, no. 28, pp. 43701-43727, November 2023.