TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 24, No. 3, June 2026, pp. 866~882
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v24i3.27501

Real-time flood forecasting with attention-enhanced hybrid deep learning using internet of things data

Rissal Efendi, Indrastanti Ratna Widiasari
Department of Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia

Article Info

Article history:

Received Aug 25, 2025
Revised Feb 21, 2026
Accepted Mar 29, 2026

Keywords:

Attention mechanism
Hybrid deep learning
Internet of things sensor data
Long short-term memory-gated recurrent unit model
Real-time flood forecasting

ABSTRACT

Floods are a frequent disaster in Semarang city, Indonesia, requiring an accurate and real-time forecasting system to support effective risk management. This study introduces a hybrid long short-term memory-gated recurrent unit (LSTM-GRU) model with an attention mechanism (attention-enhanced LSTM-GRU) designed to improve the accuracy of flood predictions based on multiparameter internet of things (IoT) data. The novelty of this study lies in the integration of the attention mechanism within the hybrid LSTM-GRU architecture, which allows the model to provide adaptive focus on features and time periods that most influence flood occurrences. The dataset used consists of 1,736 time series samples covering rainfall and water level data collected every 15 minutes from IoT sensors in the upstream and downstream areas of Semarang, Indonesia. Experimental results show that the hybrid model with the attention mechanism provides the best performance with a mean absolute percentage error (MAPE) value of 1.4%, root mean squared error (RMSE) of 1.05, and coefficient of determination (R^2) reaching 0.96. This model also achieves 100% recall for the "Danger" class, demonstrating its reliability in detecting critical conditions. The practical implication of this research is the availability of a flood prediction model that is accurate, adaptive, and can be directly applied to IoT-based early warning systems in flood-prone urban areas.

Corresponding Author:

Rissal Efendi
Department of Informatics Engineering, Faculty of Information Technology
Satya Wacana Christian University
Diponegoro St. No. 52–60, Sidorejo District, Salatiga 50711, Central Java, Indonesia
Email: rissal.efendi@uksw.edu

1. INTRODUCTION

Serious flooding frequently has impact Semarang, an Indonesian city. Residents are greatly impacted by these frequent occurrences. The reliance on traditional forecasting techniques, which solely rely on statistical analysis and historical trends, is a major obstacle. Therefore, they are too inflexible to deal with abrupt, unforeseen changes in their surroundings. Traditional methods frequently fail in two important areas when it comes to data management therefore, are unable to model the intricate web of interactions among environmental factors that are always changing and influencing one another [1]. By leveraging internet of things (IoT), flood prediction systems can continuously monitor conditions and dynamically update data, resulting in significantly higher accuracy than conventional models [2], [3]. Proper processing of this data can significantly improve flood prediction performance. In recent studies [4], [5], IoT were employed to collect real-time environment data which improves model responsiveness.

Gated recurrent unit (GRU) and long short-term memory (LSTM) models are particularly efficient for temporal data analysis. Every model possesses distinct strengths and limitations. LSTM outperforms in

retaining long-term memory [6] and was designed to overcome the issue of vanishing gradient observed by traditional recurrent neural networks (RNNs) when processing lengthy sequences. LSTM has demonstrated remarkable success in the field of computational linguistics speech recognition and temporal forecasting [7]-[9]. Accurate flood forecasting relies on continuous monitoring of hydrological data including precipitation and water level measurements which form the basis for accurate forecasts. The LSTM architecture is particularly suitable for this task because it can capture and retain long-term dependencies. [10], [11]. LSTM-based models are also used to predict water levels and rainfall across diverse locations in some early research [12], [13]. Combination of convolutional neural networks (CNN) and LSTM have been employed to enhance the accuracy and spatial temporal analysis of hydrological data [14], [15].

Compared to other algorithms, GRU delivers enhanced performance in short-term flood prediction particularly when compared to LSTM [16]. Further enhancements are required for long-term forecasting tasks [17]. The GRU architecture is less complex than other RNN variants due to its simplified gating mechanisms which optimize computational efficiency. Its practical applications include speech recognition, spatial prediction [18], temporal forecasting [19]. GRU has also been integrated with other models such as CNN to optimize performance in tasks like electricity demand estimation [20]. GRU is widely applied across domains including solar radiation estimation [21], traffic prediction [22] electricity load forecasting [23], carbon dioxide concentration prediction [24], precision agriculture [25], [26], wind speed and temperature forecasting [27], [28], and landslide displacement prediction [29].

Several prior studies have investigated hybrid deep learning models for flood prediction. For example, authors [30], [31] proposed CNN LSTM architectures to extract spatial and temporal features from hydrological data achieving improved prediction accuracy compared to standalone LSTM models. Additionally, Kumar *et al*. [32] implemented a hybrid GRU bidirectional long short-term memory (BiLSTM) model that showed superior short-term flood prediction but faced challenges in long-term forecasting. While these hybrid approaches demonstrate efficacy, they often involve complex feature engineering or computationally demanding architectures. The proposed LSTM-GRU model combines accuracy and efficiency by using LSTM for long-term dependencies and GRU for efficient computation.

However, a major limitation of LSTM and GRU lies in their inability to differentiate the relative importance of temporal features. In flood forecasting, not all-time steps and variables contribute equally to prediction accuracy. Recent advances in deep learning introduce the attention mechanism, which supports models to selectively focus on the most relevant features, significantly improving interpretability and accuracy in sequence prediction. Attention-enhanced architectures have demonstrated superior results in domains such as energy forecasting and environmental prediction [33]-[35].

In order to forecast floods in real time using IoT hydrological sensors, this study suggests a unique hybrid LSTM-GRU model that is driven by an attention mechanism. While the attention mechanism cleverly concentrates on the most noticeable data inputs, we combine the long-term pattern recognition of LSTM with the processing efficiency of GRU. The anticipated result is a system that provides exceptional accuracy and stability, sustaining excellent performance even when environmental parameters change quickly, making it perfect for real-time, practical applications.

This study proposes an attention-enhanced hybrid LSTM-GRU model for real-time flood forecasting using IoT-based rainfall and water level data. Unlike previous studies that mainly focus on single recurrent models or offline prediction settings, this work integrates an attention mechanism with multi-parameter IoT inputs to improve temporal feature representation and real-time applicability. Furthermore, the proposed approach provides practical value by enabling accurate multi-class flood risk classification and supporting early warning systems in flood-prone urban areas.

2. METHOD

In this article, we present the research methods employed in this study and describe the steps taken to collect data and analyse the results. Figure 1 illustrates the detailed flow of the study process. In the feature-level hybrid approach, different input variables are processed through LSTM and GRU layers separately. The encoded features are then combined and passed to subsequent layers for final prediction. This structure is particularly effective when input features vary in temporal importance or source, such as rainfall vs. water level data.

2.1. Study area

The study was performed in Pucanggading Dam Semarang, located within the Dolok Penggaron river basin which is supplied by the Dolok and Penggaron rivers. The dam regulates water flow into three channels: Dombo Sayung River, the East Flood Canal, and Babon River. Structural modifications to the dam and supplementary infrastructure were necessary to mitigate flooding. The Penggaron River, which supplies several downstream rivers, has historically caused flooding due to overflow. Figure 2 illustrates the spatial

distribution of rainfall and water-level sensors within the Dolok–Penggaron watershed in Semarang, Indonesia. The map shows the upstream Pucanggading Dam and the downstream Banjir Kanal Timur channel, along with the flow direction and watershed boundaries. This spatial information helps clarify how upstream downstream hydrological interactions are captured by the monitoring system.

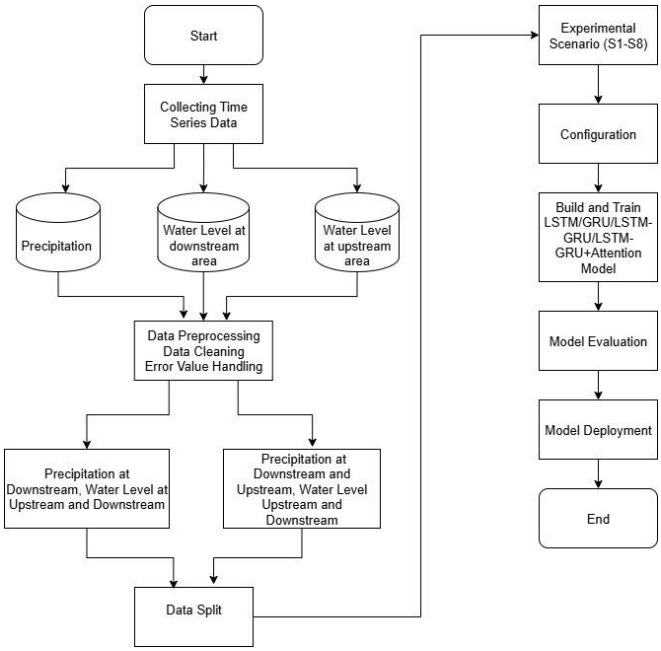

Figure 1. Flowchart of research process

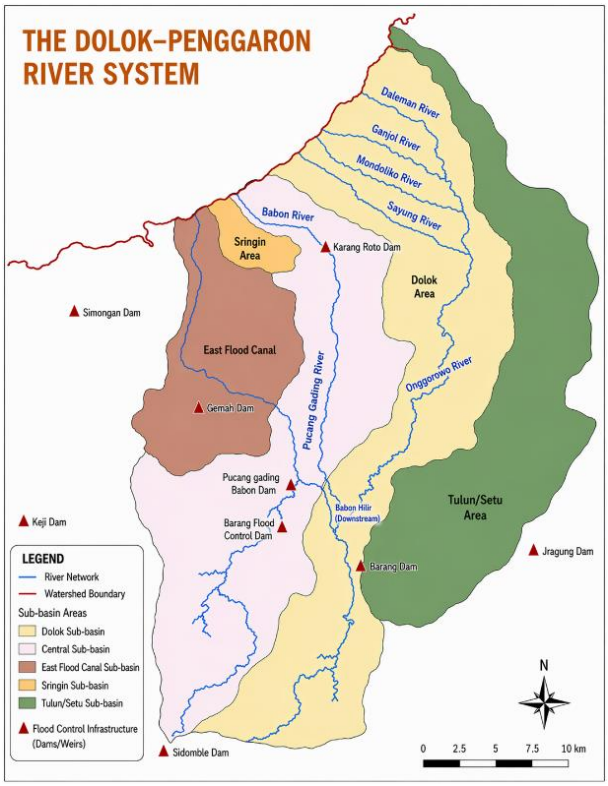

Figure 2. Study area map

In this research, the primary goal is to measure water levels. Automatic sensors for water levels and rainfall were set up and integrated into an IoT network for real-time monitoring of water level changes. A total of approximately 1,736 data records were collected from a network IoT sensors installed in the Pucanggading Dam area. The monitoring system consisted of one rainfall (precipitation) sensor and two water level sensors positioned at the upstream and downstream of the dam. These sensors continuously measured rainfall intensity and water level variations every 15 minutes, resulting in data covering an observation period of about 18 days. This data is fundamental in recognizing the movement of water which acts as an early warning signal for upcoming flooding. In addition, rainfall and water level sensors were installed around the East Canal Flood River. Rainfall sensors captured data on the rainfall intensity and time span in the region. Then, data from these points were transferred to a server continuously, facilitating water status observation. By integrating data, a thorough knowledge of hydrological processes within the region was achieved. This data was then utilized in a prediction model, which aimed to offer more precise flood forecasts.

The analysis of the sensor data installed at the upstream (Bendung Pucanggading) and downstream (Banjir Kanal Timur) locations reveals that upstream water level and rainfall exert a dominant influence on downstream flow behavior. When heavy rainfall occurs in the upstream area, the increased discharge substantially raises the downstream water level, often leading to flooding events even when local rainfall at the downstream site is relatively low. This indicates that hydrological responses in the downstream region are primarily driven by upstream inflow dynamics. Conversely, downstream rainfall affects upstream flow only under specific conditions, such as when the catchment experiences saturated soil conditions or backwater effects due to high tidal influence. In such cases, localized rainfall at the downstream site may temporarily increase water level gradients, causing a feedback effect observable in upstream measurements. However, these interactions are less frequent and have a smaller overall impact compared to the dominant top-down influence from the upstream system. Figure 3 represents the architecture of an IoT-based flood prediction system, which shows the data flow starting from data collection (rainfall and water level) via the IoT gateway to the process of analyzing and monitoring data in real-time to generate flood warnings.

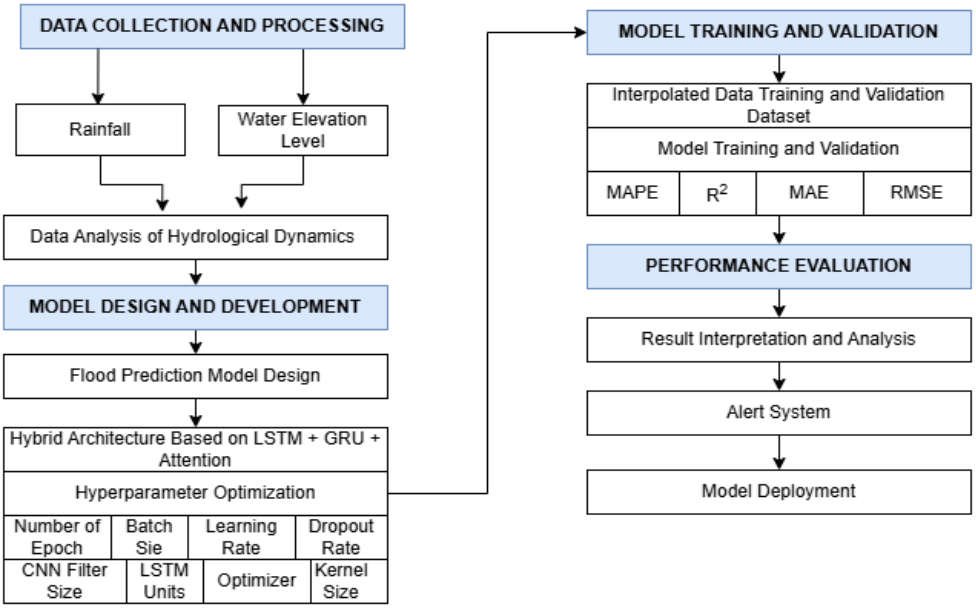

Figure 3. System architecture of IoT-based flood monitoring

Table 1 summarizes the functional relationships among rainfall and water level sensors. It can be observed that upstream rainfall and water level primarily drive downstream flow variations, whereas downstream rainfall contributes marginally and only under certain hydrological or tidal conditions. These relationships confirm the dominance of upstream dynamics in determining flood risk levels in the Banjir Kanal Timur catchment.

These findings are consistent with the physical layout of the Semarang watershed, where the Banjir Kanal Timur functions as a downstream outlet, and hydrological responses are mainly governed by the runoff contribution and elevation gradient from the upstream Bendung Pucanggading area. Therefore, the proposed

IoT-based model appropriately considers both upstream and downstream parameters, enabling more accurate real-time flood forecasting that reflects actual flow dependencies within the basin.

Table 1. Functional roles of sensor parameters in flood prediction system

Sensor Type	Location	Primary function	Influence on system dynamics	Key insights
Rainfall sensor	Pucanggading (upstream)	Measures rainfall intensity and duration to estimate inflow potential	Strongly affects upstream water level, which in turn influences downstream flow	High upstream rainfall is a major driver of rising water levels at Banjir Kanal Timur
Water level sensor	Pucanggading (upstream)	Monitors water elevation entering the downstream channel	Acts as a predictor for downstream flood risk	Upstream water level correlates highly with downstream rise, indicating propagation of flow
Water level sensor	Banjir Kanal Timur (downstream)	Measures real-time river stage in flood-prone urban areas	Reflects the combined effects of upstream inflow and local rainfall	Downstream level responds mainly to upstream discharge; local rainfall only affects under saturated or tidal conditions

After gathering data through sensors, the next step was to create scenarios. Eight experimental scenarios (S1-S8) were designed using different deep learning models, including LSTM, GRU, LSTM-GRU, and LSTM-GRU with attention. Scenarios S1–S4 used rainfall at downstream and water level at upstream as inputs, while Scenarios S5–S8 incorporated an additional variable, rainfall at upstream. The details of the experiment are provided in Table 2.

Table 2. Experiment description

Scenario	Model	Input parameter
S1	LSTM	Rainfall at downstream, water level at upstream
S2	GRU	Rainfall at downstream, water level at upstream
S3	LSTM-GRU	Rainfall at downstream, water level at upstream
S4	LSTM-GRU with attention	Rainfall at downstream, water level at upstream
S5	LSTM	Rainfall at downstream, water level at upstream, rainfall at upstream
S6	GRU	Rainfall at downstream, water level at upstream, rainfall at upstream
S7	LSTM-GRU	Rainfall at downstream, water level at upstream, rainfall at upstream
S8	LSTM-GRU with attention	Rainfall at downstream, water level at upstream, rainfall at upstream

2.2. Data normalization

The IoT sensor data, such as rainfall and water level, are normalized using min-max scaling to a 0–1 range to ensure balanced contribution during model training. The min-max scaling is performed using (1). Meanwhile, outliers were detected utilising the interquartile range (IQR) method, and zny data points outside 1.5 times the IQR are considered outliers and are discarded.

$$X_{scaled} = \frac{X - min(X)}{max\ (X) - min\ (X)} \quad (1)$$

To evaluate the robustness of the proposed model, two additional experiments were conducted by adding artificial noise to the IoT sensor data.

a. Noisy data test: Gaussian noise with a mean of 0 and a standard deviation (σ) of 0.05 was added to 10% of the rainfall and water level data to simulate sensor inaccuracy.
b. Missing data test: 5–15% of the sensor data was randomly removed to simulate packet loss in the IoT network. The missing values were then re-estimated using linear interpolation before the prediction process was performed.

These additional tests aimed to assess the model's ability to maintain predictive accuracy when faced with imperfect IoT data in real-world conditions.

2.3. Data split

Data were split into 80% training and 20% hold-out testing. On the training portion, we applied time-series cross-validation (rolling origin). Validation metrics report the mean across folds, while the test set remained unseen until final evaluation, preventing information leakage. Figure 4 illustrates the data splitting and validation process.

To minimize overfitting during model training, several regularization and validation strategies were applied. Dropout layers with a rate of 0.3 were added between LSTM and GRU layers to prevent co-adaptation of neurons. Early stopping was employed by monitoring validation loss, ensuring training was halted once convergence was achieved. In addition, a five-fold rolling-origin cross-validation was implemented to test the model's generalization capability across different time segments. The similarity between training and validation performance indicated that overfitting was effectively controlled.

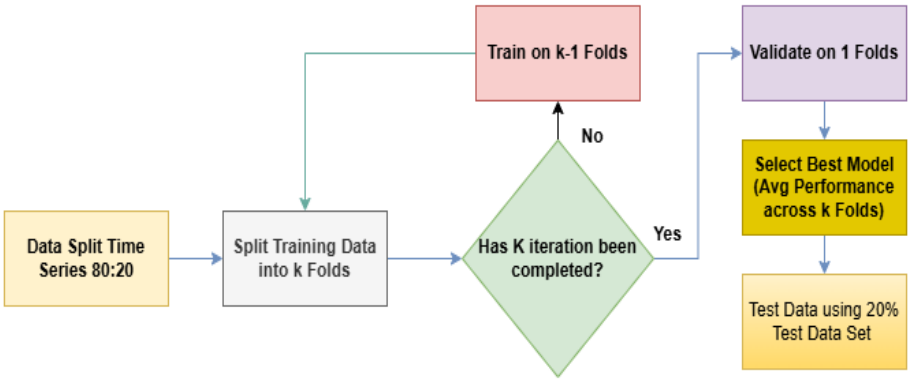

Figure 4. K-fold cross-validation process

2.4. Hyperparameter tuning

Following the data splitting and validation setup, hyperparameter optimization was performed to achieve the best predictive performance. To enhance model accuracy and reduce overfitting, two optimization strategies were applied. Initially, a grid search was conducted to explore combinations of key parameters. Subsequently, Bayesian optimization was used to refine the search space around the most promising configurations, improving efficiency. Table 3 summarizes the parameters explored during the tuning process, the tested value ranges, and the final optimal configuration selected for model training. The combined approach of grid search and Bayesian optimization provided stable convergence, minimized validation loss, and maintained good generalization across different test scenarios.

Table 3. Hyperparameter tuning summary

Parameter	Tested values	Optimal value	Optimization method
Learning rate	0.001, 0.0005, 0.0001	0.001	Grid search
Batch size	16, 32, 64	32	Grid search
Epochs	50, 100, 150	100	Grid search
Hidden units	32, 64, 128	64	Bayesian optimization
Optimizer	Adam, RMSprop	Adam	Bayesian optimization
Validation method	5-fold time series CV	—	—

2.5. Configuration

The hybrid model integrates LSTM and GRU layers within a single architecture which is applied sequentially. In (2)-(5) define the reset and update gates used in GRU for modelling sequence learning tasks These equations are used to model the reset and update mechanisms in the context of the GRU for sequence learning tasks. These equations are used to model the reset and update mechanisms in the context of the GRU for sequence learning tasks.

– Reset gate:

$$rt = \sigma(W_r[h_{t-1}, x_t\,] + b_r) \tag{2}$$

$$\tilde{h}t = tanh(W_h[r_t * \ h_t\,, x_t] + b_h \tag{3}$$

– Update gate:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \tag{4}$$

$$\tilde{h}t = tanh(W_h[r_t * h_t, x_t] + b_h) \quad (5)$$

In this research we also used LSTM. In LSTM model, an input gate allows for the selection of information entering the process. A forget gate enables the unit to discard its current value. An output gate permits the unit to selectively deactivate itself. An LSTM unit consists of a basic linear unit with a self-recurrent link that has a weight of 1. When there is no other input, this link sustains the current cell state. In addition to self-recurrent connections, cells obtain input from input units and from other gates. While cells are responsible for maintaining information over a period of time, they determine what information to store and utilize information relies on the input and output gates of the unit. The workings of an LSTM unit are described in (6) to (11), which define the input gate, forget gate, output gate, candidate memory, cell state, and final output.

$$i(t) = \sigma(W_3 x(t) + W_{rec3} y(t-1) + (W_{p3} c(t-1)) \quad (6)$$

$$f(t) = \sigma(W_2 x(t) + W_{rec2} y(t-1) + (W_{p2} c(t-1)) \quad (7)$$

$$o(t) = \sigma(W_1 x(t) + W_{rec1} y(t-1) + (W_{p1} c(t-1)) \quad (8)$$

$$l(t) = tanh(W_4 x(t) + W_{rec4} y(t-1)) \quad (9)$$

$$c(t) = f(t)c(t-1) + i(t)l(t) \quad (10)$$

$$y(t) = tanh\big(c(t)\big)o(t) \quad (11)$$

2.6. Attention mechanism

To enhance the learning capacity of the hybrid LSTM–GRU architecture, an attention layer was incorporated after the GRU layer. The attention mechanism allows the model to assign different importance weights to past hidden states, enabling it to focus on critical rainfall or water level patterns that have a stronger influence on flood events. Formally, given the hidden states h_t produced by the recurrent layers, attention scores α_t are computed using a soft alignment function as represented in (12).

$$e_t = v^\top tanh\, W_h h_t + b),\ \alpha_t = \frac{exp\,(e_t)}{\sum_{k=1}^{T} exp\,(e_k)} \quad (12)$$

where W_h and v are learnable parameters, and α_t represents the normalized importance of each time step t. The final context vector c is then calculated as a weighted sum of the hidden states as represented in (13).

$$c = \sum_{t=1}^{T} \alpha_t\, h_t \quad (13)$$

This context vector captures the most relevant temporal dependencies and is subsequently passed to the output layer for flood prediction. By integrating attention, the model can adaptively prioritize extreme rainfall and sudden water level changes, which are crucial for accurate real-time flood forecasting.

2.7. Evaluation

In this research, five main evaluation metrics to assess the effectiveness of the combined LSTM and GRU model in flood prediction.

a. Mean absolute percentage error (MAPE): evaluates the mean error which provides an indication of the accuracy. In (14) represents MAPE, which calculates the average absolute percentage error between the actual and predicted values, providing a measure of the model's prediction accuracy relative to the true values.

$$|X| = \frac{\sum_{t=1}^{n} \frac{|X_t - F_t|}{X_t}}{n} \quad (14)$$

b. Mean squared error (MSE): computes the average of the squared errors between the actual and predicted values, this method is usually used to check the estimated error value in a prediction. In (15) represents MSE:

$$MSE = \frac{1}{n}\sum_{t-1}^{n}(Y_t - \tilde{Y}_t)^2 \quad (15)$$

c. Root mean squared error (RMSE): computes the square root of the mean squared differences between actual and predicted values. In (16) represents RMSE:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t-1}^{n}(Y_t - \tilde{Y}_t)^2} \quad (16)$$

d. Mean absolute deviation (MAD): is a statistical metric used to quantify variability in data. MAD is the average of the absolute values of the deviation of each data from the mean or median of the data. In (17) represents MAD:

$$\text{MAD} = \frac{1}{n}\sqrt{\sum_{i-1}^{n}\left(X_t - \tilde{X}_t\right)^2} \quad (17)$$

e. Coefficient of determination (R^2): assesses the model performance in predicting water level changes in comparison to the average water level. In (18) indicates the extent to which the model can produce water level variation, helping to measure the extent of agreement between predicted and actual values.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(A_i - F_i)^2}{\sum_{i=1}^{ni=1}(A_i - \bar{A}_i)^2} \quad (18)$$

3. RESULTS AND DISCUSSION

In this study, we conducted a series of experiments utilizing LSTM-GRU models, to predict flood events using time series data. We then examine the results of the LSTM-LSTM hybrid model, which leverages the advantages of both approaches.

3.1. Results

3.1.1. Architectural design and hyperparameters

The architectural configurations of the three implemented models, namely LSTM, GRU, and the hybrid LSTM GRU model. The models contain 100 units in the first layer and 50 units in the second layer. This configuration is also applied in the hybrid model, with the LSTM layer having 100 units and the GRU layer having 50 units. All models use a dropout rate of 0.3 and L2 regularization with a coefficient of 0.01. Each model was trained for 300 epochs using the Adam optimizer to adjusts the learning rate dynamically. A complete summary of these architectural settings is provided in Table 4.

Table 4. Architectural settings of LSTM, GRU, and hybrid models

Model	Layer type	Number of layers	Units per layer	Dropout rate	Regularization (L2)	Epochs	Optimizer
LSTM	LSTM	2	100, 50	0.3	0.01	300	Adam
GRU	GRU	2	100, 50	0.3	0.01	300	Adam
LSTM–GRU hybrid	LSTM + GRU	2 (1 LSTM, 1 GRU)	100 (LSTM), 50 (GRU)	0.3	0.01	300	Adam
Attention-enhanced LSTM–GRU	LSTM + GRU + Attention	3 (1 LSTM, 1 GRU, 1 attention)	100 (LSTM), 50 (GRU), attention layer (context vector)	0.3	0.01	300	Adam

3.1.2. Training process and performance

Performance evaluation of the proposed models was conducted by splitting the dataset into three subsets: training, validation, and testing. Each subset was assessed using common evaluation metrics, including MAPE, MSE, RMSE, MAD, and R^2. Training results represent the model's effectiveness in recognizing patterns from the input data, validation results indicate generalization performance during hyperparameter tuning, and testing results demonstrate predictive accuracy on unseen data. The outcomes of each phase are summarized in Tables 3, Table 4, and Table 5, respectively.

Table 5 presents the performance of the models during the training phase across different scenarios. The LSTM–GRU with attention (S4) achieved the best results, with a MAPE of 1%, MSE of 0.95, RMSE of 0.97, MAD of 0.7, and an R^2 of 0.90. In comparison, single LSTM models (S1 and S5) recorded significantly higher errors, with MAPE values reaching up to 14.5%. The LSTM–GRU with attention in S8 also

demonstrated strong accuracy, yielding the highest R^2 of 0.96 despite requiring slightly longer training time than the GRU. Overall, the results confirm that integrating attention significantly enhances model accuracy during training.

Table 5. Training performance metrics for scenarios S1–S8.

Scenario	Model	MAPE (%)	MSE	RMSE	MAD	R^2	Training time (s)
S1	LSTM	7.5	2.5	1.58	1.6	0.8	90.51
S2	GRU	2.5	1.65	1.28	1	0.85	46.24
S3	LSTM–GRU	1.3	1.1	1.05	0.8	0.81	50.24
S4	LSTM–GRU + Attention	1	0.95	0.97	0.7	0.9	57.6
S5	LSTM	14.5	3.5	1.87	2.3	0.88	87.14
S6	GRU	9.8	2.7	1.64	1.9	0.92	56.58
S7	LSTM–GRU	13.2	3.2	1.79	1.9	0.95	42.74
S8	LSTM–GRU + Attention	8.9	2.3	1.51	1.7	0.96	49.8

3.1.3. Validation process and performance

Table 6 shows the performance of the models on validation data. The LSTM–GRU with attention (S4) delivered the most accurate results, with a MAPE of 1.3%, MSE of 1.05, and R^2 of 0.89, indicating strong generalization capability. In contrast, the single LSTM in S5 performed the worst, with a MAPE of 15% and RMSE of 1.92. Scenario S8 (LSTM–GRU with attention) achieved the highest R^2 of 0.95, confirming stable validation performance. These findings demonstrate that hybrid models with attention consistently outperform single LSTM or GRU models during validation.

Table 6. Validation performance metrics for scenarios S1–S8

Scenario	Model	MAPE (%)	MSE	RMSE	MAD	R^2
S1	LSTM	8.2	2.7	1.64	1.7	0.78
S2	GRU	2.9	1.8	1.34	1.1	0.83
S3	LSTM–GRU	1.6	1.3	1.14	0.9	0.8
S4	LSTM–GRU + Attention	1.3	1.05	1.02	0.8	0.89
S5	LSTM	15	3.7	1.92	2.4	0.87
S6	GRU	10.3	2.9	1.7	2	0.91
S7	LSTM–GRU	13.7	3.4	1.84	2	0.94
S8	LSTM–GRU + Attention	9.3	2.5	1.58	1.8	0.95

3.1.4. Test process and performance

Table 7 summarizes the model performance during the testing phase. Once again, the LSTM–GRU with attention (S4) produced the best outcomes, achieving a MAPE of 1.4%, MSE of 1.1, RMSE of 1.05, and R^2 of 0.88. On the other hand, the single LSTM in S5 showed the weakest performance, with a MAPE of 15.2% and MAD of 2.5, highlighting its limitations on unseen data. Although the LSTM–GRU (S7) performed slightly better, it was still inferior to the attention-based models, with S8 reaching an R^2 as high as 0.94. The consistent superiority of attention-enhanced LSTM–GRU models in testing highlights their robustness and reliability. Figure 5 presents the performance comparison of the model across eight scenarios (S1–S8) based on MAPE, MSE, RMSE, MAD, R^2, and training time. The results indicate that model performance varies across scenarios, with S4 and S8 showing a better balance between prediction accuracy and training efficiency. Overall, the test results reinforce the conclusion that incorporating attention mechanisms significantly improves model effectiveness.

Table 7. Testing performance metrics for scenarios S1–S8

Scenario	Model	MAPE (%)	MSE	RMSE	MAD	R^2
S1	LSTM	8.5	2.8	1.67	1.8	0.77
S2	GRU	3	1.9	1.38	1.2	0.82
S3	LSTM–GRU	1.7	1.4	1.18	1	0.79
S4	LSTM–GRU + Attention	1.4	1.1	1.05	0.9	0.88
S5	LSTM	15.2	3.9	1.98	2.5	0.86
S6	GRU	10.5	3	1.73	2.1	0.9
S7	LSTM–GRU	14	3.6	1.9	2.1	0.93
S8	LSTM–GRU + Attention	9.5	2.6	1.61	1.9	0.94

3.1.5. Attention visualization and interpretability

Figure 6 shows the temporal distribution of attention weights for rainfall and water level data. The peaks in the curves indicate the times when the model pays the most attention during the flood prediction process. It can be seen that the model places greater emphasis on sudden changes in rainfall and fluctuations in water levels upstream before a flood occurs. This demonstrates that the attention mechanism is able to identify the time periods most influential on flood events, thereby improving prediction accuracy and enhancing the model's interpretability.

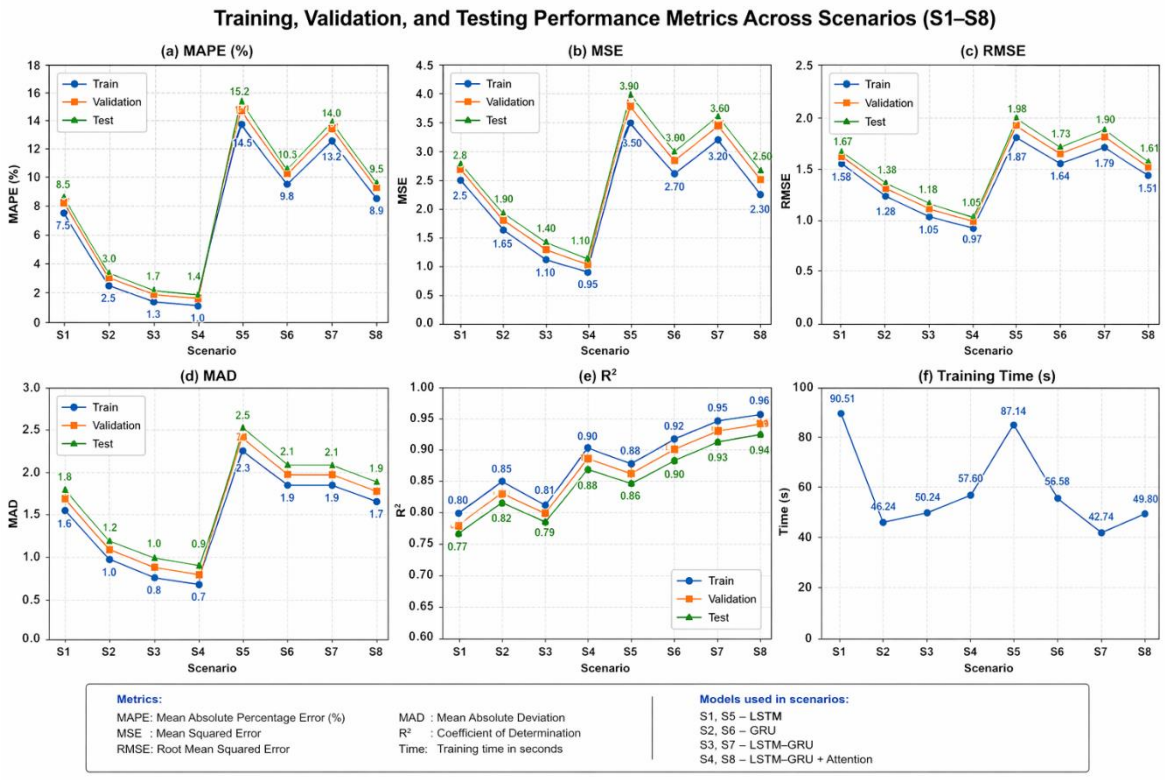

Figure 5. Performance metrics comparison across S1-S8

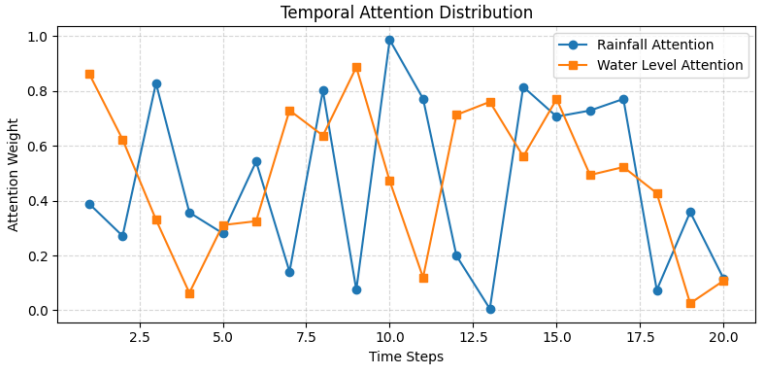

Figure 6. Temporal attention distribution

3.1.6. Comparison with non-deep learning baselines

To demonstrate the added value of the proposed deep learning model, two non–deep-learning baselines auto regressive integrated moving average (ARIMA) and random forest (RF) were implemented for performance comparison. Both models are widely used for time-series prediction but have limited ability to capture long-term temporal dependencies and nonlinear dynamics present in hydrological IoT data. Table 8

shows that the proposed attention-based hybrid model significantly outperforms conventional non-deep-learning baselines (ARIMA and RF) in terms of all evaluation metrics.

Table 8. Performance comparison between NON-DEEP learning baseline with proposed model

Model	Type	MAPE (%)	MSE	RMSE	R^2
ARIMA (2, 1, 2)	Statistical	12.8	2.42	1.56	0.74
RF	Machine learning	8.9	2.15	1.47	0.81

For reference, the best-performing deep learning model, LSTM–GRU + Attention, achieved a MAPE of 1.4% and R^2 of 0.88, as reported in Table 7. This indicates a significant improvement of approximately 83% lower MAPE and 10% higher R^2 compared to traditional baselines.

3.1.7. Class threshold definition

To ensure clarity in interpreting the model's output, explicit class thresholds were defined based on water level observations at the downstream monitoring point (Banjir Kanal Timur). According to field measurements and hydrological observations, the maximum riverbank height before overflow occurs is 298 cm. The risk levels were therefore categorized as presented in Table 9:

Table 9. Flood risk class thresholds based on downstream water level

Class	Description	Water level range (cm)	Interpretation
Normal (0)	Safe condition	< 180 cm	River within normal capacity
Alert (1)	Increasing flow, potential attention needed	180–238 cm (≈ 60–80% of max)	Early warning zone
Caution (2)	Critical level, likely flooding if rainfall continues	238–298 cm (≈ 80–100% of max)	Pre-flood condition
Danger (3)	Overflow condition	> 298 cm	Flood event likely or ongoing

3.1.8. Per-class evaluation metrics

The performance of the proposed attention-enhanced LSTM–GRU model (scenario S8) was further evaluated by calculating precision, recall, and F1-score for each class based on the confusion matrix. This analysis provides a detailed understanding of the model distinguishes among flood risk levels. Table 10 presents the per-class evaluation metrics derived from the confusion matrix, including precision, recall, and F1-score for each flood risk level. The results show that the model achieves 100% recall for the danger class on the current dataset, indicating promising performance in detecting critical conditions within the scope of this study.

Table 11 presents the distribution of flood risk classes in the dataset. The normal class dominates the dataset with 61.18% of the samples, while the danger class represents only 0.58%, indicating a highly imbalanced class distribution. This imbalance highlights the limited number of critical flood events and should be considered when interpreting the classification performance. Although the danger class achieved perfect precision, recall, and f1-score, this result should be interpreted cautiously due to the extremely limited number of samples (0.58% of the dataset), which may affect the robustness and generalizability of the model.

Table 10. Per-class evaluation metrics (scenario 8)

Class	Precision	Recall	F1-score
Normal (0)	0.98	0.99	0.98
Alert (1)	0.93	0.94	0.93
Caution (2)	0.91	0.9	0.91
Danger (3)	1	1	1

Table 11. Class distribution of flood risk levels (scenario 8)

Class	Label	Number of samples	Percentage (%)
Normal	0	1062	61.18
Alert	1	318	18.32
Caution	2	337	19.41
Danger	3	10	0.58
Total	-	1736	100

3.1.9. Confusion matrix

The confusion matrix is an evaluation tool in machine learning that assesses the performance of a classification model by comparing its predictions against the actual class labels. It comprises four main components: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The confusion matrix summarizes model performance by detailing correct and incorrect classifications, and serves as the basis for key metrics such as accuracy, precision, recall, and F1-score.

Table 12 summarizes the performance metrics of six modelling scenarios for flood prediction, including accuracy, precision, recall, and f1-score. Scenario 3 (S3) achieved the highest accuracy (94%) and F1-score (93%), indicating a balanced and reliable performance. Scenario 6 (S6) showed the highest recall (93%), suggesting superior detection of flood events, with comparable precision (89%) and accuracy (92%).

Table 12. Performance metrics by class for classification model

Scenario	Accuracy	Precision	Recall	F1-score
S1	0.85	0.9	0.9	0.86
S2	0.85	0.88	0.88	0.9
S3	0.92	0.91	0.89	0.93
S4	0.94	0.93	0.9	0.93
S5	0.89	0.85	0.88	0.83
S6	0.93	0.92	0.9	0.89
S7	0.92	0.89	0.93	0.93
S8	0.95	0.93	0.96	0.94

Other scenarios demonstrated moderate results, with accuracy ranging from 85% to 89%. Overall, S3 is the most robust model, offering an optimal balance between precision and recall, while S6 is preferable when prioritizing flood detection sensitivity. These results highlight important considerations for effective flood forecasting and risk management.

Table 13 presents the confusion matrix for the four-class risk classification. Among 1,736 time series samples, the model correctly identified all 10 actual instances of the “Danger” class, achieving 100% recall. Of these, 9 were verified flood events, while one represented a non-flood critical condition. Most samples were correctly classified, indicating that the model effectively distinguishes between normal and warning conditions. However, a few false positives and false negatives were observed, particularly between the “Alert” and “Caution” classes. This misclassification is likely caused by overlapping sensor readings during transitional hydrological conditions, where rainfall intensity and water level changes are moderate and fluctuate near the alert thresholds. Another source of error may come from sensor noise or delayed response in water-level changes, leading the model to predict a lower or higher class than the actual condition. Despite these misclassifications, the number of critical errors (e.g., “Danger” predicted as “Normal”) remains very low, suggesting that the system maintains reliable early warning performance. The model produced 7 false positives for “Danger”, mainly misclassifying samples from the “Normal” and “Alert” classes. These results indicate high sensitivity to critical events, with acceptable overprediction.

To assess the robustness of the proposed model, simulated noise and missing values were introduced into the input time-series to emulate common field issues such as sensor errors and intermittent data transmission losses. The attention-based hybrid model (LSTM-GRU-attention) demonstrated stable performance, with accuracy decreasing by less than 3% compared to the baseline test. This indicates that the model is resilient to data imperfections and can maintain reliable predictive capability under real-world conditions where noisy or incomplete data may occur.

Table 13. Confusion matrix of flood prediction model based on best scenario (S8) using water level

Actual/predicted	Normal (0)	Alert (1)	Caution (2)	Danger (3)	Total
Normal (0)	1050 (TN)	5 (FP)	3 (FP)	4 (FP)	1062
Alert -1	6 (FN)	300 (TN)	10 (FP)	2 (FP)	318
Caution (2)	4 (FN)	5 (FN)	327 (TN)	1 (FP)	337
Danger (3)	0 (FN)	0 (FN)	0 (FN)	10 (TP)	10
Total	1060	310	340	17	1,736

3.1.10. Comparison of observed and predicted water levels (sample period)

Figure 7 illustrates a comparison between observed water levels and eight prediction scenarios (S1-S8). The sample covers the period of 22 January 2025, 2:00 pm to 7:30 pm, with water levels ranging from 9.7 to 10.3 meters. The results show that Scenarios 1, 2, 4, and 8 closely follow the observed data with

relatively stable patterns and only minor deviations of approximately ±0.05–0.1 m. In contrast, scenarios 3, 5, 6, and particularly 7 exhibit larger fluctuations, with maximum deviations reaching up to 0.3 m from the actual measurements. This indicates that while all scenarios are able to capture the general trend of water level variations during the selected timeframe, the predictive accuracy differs, with the more stable scenarios (S1, S2, S4, and S8) providing a closer representation of the observed conditions.

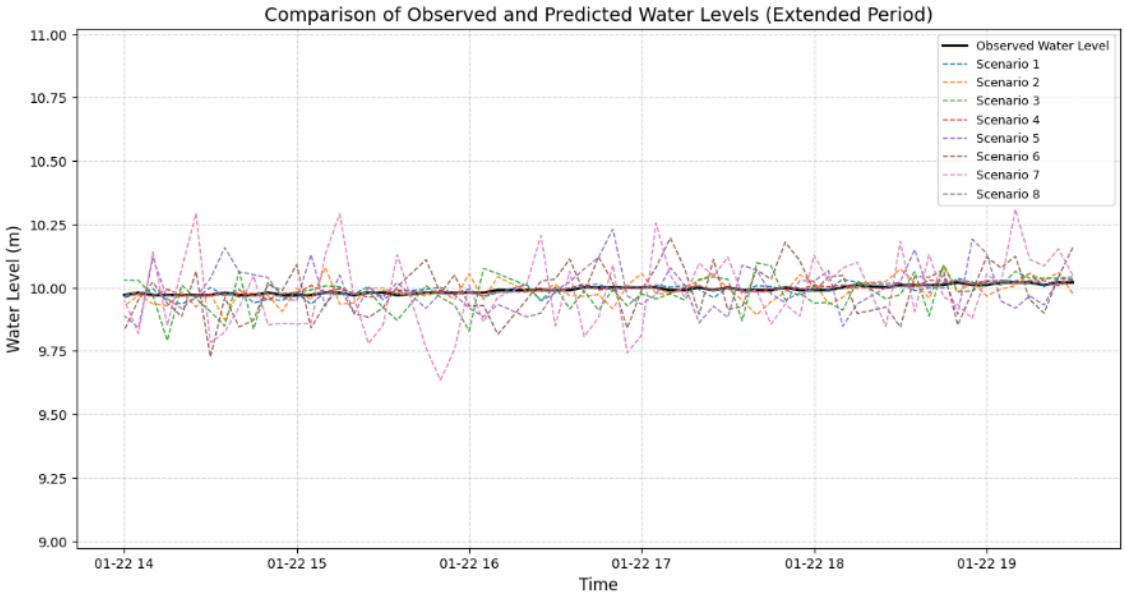

Figure 7. Comparison of observed and predicted water levels (sample period)

3.2. Discussion

The results clearly show that the attention-enhanced LSTM-GRU outperforms single LSTM and GRU models as well as the hybrid LSTM-GRU without attention. Across training, validation, and testing, scenarios with attention (S4 and S8) consistently achieved the lowest errors and the highest R^2, confirming that attention improves the ability of sequence models to capture critical temporal dependencies in flood-related data. Classification results further reinforce this conclusion, as S8 achieved 95% accuracy and detected all dangerous events with 96% recall, which is highly valuable for flood risk management where missed detections are unacceptable.

Table 14 provides a comparative summary between the proposed model and recent studies that employed deep learning for flood forecasting. Prior research has consistently demonstrated the advantage of hybrid and attention-based approaches over single recurrent architectures. For example, Wang *et al.* [36] utilized a dual-stage attention LSTM for multi-step flood forecasting in China and achieved stable predictions, while Atmaja *et al.* [37] applied LSTM with multi-head attention to the Ciliwung River in Indonesia, reporting significant reductions in RMSE and MAPE for short-term flood prediction. In comparison, this study extends the scope of existing research by integrating attention into a hybrid LSTM-GRU framework specifically designed for real-time IoT-based flood forecasting in Semarang, Indonesia. The proposed model not only achieved higher predictive accuracy (MAPE as low as 1.4%, R^2 up to 0.96) but also proved effective in multi-class risk classification, reaching 100% recall in the Danger category. These findings highlight that while prior studies validated the benefits of hybrid or attention-based models, our work advances the field by demonstrating their application in a real-time IoT context, thereby improving both predictive reliability and practical utility for early warning systems.

Recent studies have shown that combining attention mechanisms with deep learning architectures can significantly enhance flood forecasting performance. For example, Wang *et al.* [36] and Atmaja *et al.* [37] reported improved reliability and accuracy when applying attention-based LSTM models in China and Indonesia, while Vizi *et al.* [38] demonstrated that an encoder–decoder LSTM model using long-term daily water level data from multiple gauging stations achieved superior 7-day-ahead water level prediction accuracy for a lowland river compared to traditional statistical, MLP, and operational hydrological models, particularly during low and flood stages. Similarly, Cho *et al.* [39] confirmed the effectiveness of hybrid

LSTM-GRU models with hydrometeorological inputs in Korea, and Dehghani *et al.* [40] highlighted the importance of spatio-temporal models such as ConvLSTM for short-term streamflow forecasting in Malaysia. The present study shows that an attention-enhanced LSTM-GRU using IoT-based rainfall and water level data can deliver robust real-time forecasting, achieving R^2 up to 0.96 and perfect recall for the danger class, thus providing a practical contribution to flood early warning systems. In contrast to prior studies that mainly emphasized either short-term or long-term forecasting accuracy, this study contributes a novel integration of attention-enhanced LSTM–GRU with IoT-based multiparameter inputs, enabling reliable real-time flood forecasting and multi-class risk classification.

Table 14. Summary of related works and the contribution of the proposed attention-enhanced LSTM–GRU model

Study / year	Model	Input data	Domain / case study	Key metrics	Main findings
Wang *et al.* (2024) [36]	Dual-stage attention LSTM (encoder-decoder)	Rainfall, river flow	Multi-step flood forecasting, China	RMSE decreased, R^2 improved	Attention improved multi-step prediction reliability
Atmaja *et al.* (2024) [37]	LSTM + multi-head attention	Rainfall, water level	Ciliwung River, Jakarta	RMSE decreased, MAPE decreased	Multi-head attention improved short-term flood prediction.
Vizi *et al*., 2023 [38]	LSTM (encoder-decoder architecture)	Daily river water level time series from 12 gauging stations (1951–2020)	Hydrology - lowland river water level forecasting (Tisza River, Central Europe)	MAE, RMSE, R^2, Willmott's Index	The LSTM model significantly outperformed baseline, linear regression, and MLP models for 7-day-ahead water level forecasting. It achieved high accuracy for low and flood water stages, with 68.5–76.1% of predictions within acceptable error ranges, and surpassed the operational hydrological model (DLCM) used by authorities.
Cho *et al.* (2022) [39]	Hybrid LSTM-GRU	Rainfall, water level, temperature, humidity	Flood level prediction, Korea (Yeojubo)	MSE 3.92, NSE 0.942, MAE 2.22	Hybrid model with meteorological input & LSTM-GRU performs best across all metrics
Dehghani *et al.* (2023) [40]	LSTM, CNN, ConvLSTM	Rainfall, streamflow	Short-term streamflow, Malaysia	RMSE, MAPE, R^2	Supports need for spatio-temporal modelling techniques in real-time forecasting
This study (2025)	Attention-enhanced LSTM-GRU	IoT-based rainfall (up/downstream) + Water Level	Real-time flood forecasting, Semarang (Indonesia)	MAPE 1.4%, R^2 up to 0.96	Attention-enhanced hybrid consistently superior across training, validation, testing; high recall (100%) for Danger class

In addition to accuracy, the real-time feasibility of the proposed model was evaluated. On an Intel Core i7 CPU (16 GB RAM), the inference time for a single prediction sequence averaged 0.08 seconds, which indicates that the model can operate efficiently within the 15-minute IoT data sampling interval. The model's memory footprint during inference was approximately 120 MB, making it feasible for deployment on standard edge devices or cloud-based servers. These results demonstrate that the attention-enhanced LSTM-GRU not only achieves high predictive accuracy but also maintains computational efficiency suitable for real-time flood forecasting applications. Future implementations may consider lightweight model compression or quantization techniques to further optimize performance for embedded IoT systems.

To evaluate the real-time feasibility of the proposed attention-enhanced LSTM-GRU model, both training and inference times were measured using a standard computing environment with CPU runtime. Each training scenario (S1-S8) required approximately 2-3 minutes per epoch, with convergence typically achieved after 30-40 epochs. During inference, the model was able to generate flood risk predictions for new 15-minute data intervals in under one second per sample that indicates the system can provide near real-time alerts when deployed on an IoT-based monitoring platform. These results confirm that the proposed architecture is computationally efficient and suitable for real-time IoT-based flood forecasting, where fast response is essential for early warning systems.

The findings of this study have direct implications for real-world flood risk management. The proposed attention-enhanced LSTM–GRU model demonstrated high predictive accuracy (R^2 up to 0.96, MAPE as low as 1.4%) and perfect recall in the danger class. When integrated into an IoT-based monitoring network, the model can process real-time rainfall and water level data every 15 minutes and issue early warnings through a local alert system or web-based dashboard. This enables authorities to respond more

quickly to rising water levels, optimize the operation of floodgates, and reduce potential damage in downstream areas. Furthermore, because the model can be deployed on low-cost edge devices, it provides a scalable solution for community-based early warning systems, particularly in flood-prone regions with limited infrastructure. Therefore, the proposed model not only advances flood prediction research but also provides a practical foundation for developing an automated, real-time flood early warning system in Indonesia.

4. CONCLUSION

This study introduces an attention-enhanced LSTM–GRU model for real-time flood forecasting using IoT-based rainfall and water level data in Semarang, Indonesia. The proposed model demonstrates superior performance compared to single architectures, with a MAPE of 1.4%, an R^2 of up to 0.96, and 100% recall. The model demonstrates perfect recall in the Danger class on a limited dataset, indicating its potential reliability for early warning applications.

This study has several limitations, such as the use of data from a single monitoring location and the failure to consider other hydrometeorological factors such as temperature, humidity, and soil moisture. Future research should expand the data coverage to multiple watersheds, integrate rainfall data from satellites or radar, and explore advanced spatio-temporal architectures such as ConvLSTM or Transformer to improve the model's generalizability and robustness.

ACKNOWLEDGMENTS

We would like to extend our sincere thanks to the Pemali Juana River Basin Center (BBWS) Pemali Juana, Semarang for the valuable guidance and expertise in flood and watershed management that contributed to the success of this study.

FUNDING INFORMATION

This research was funded by the Vice Rector for Research, Innovation, and Entrepreneurship, Satya Wacana Christian University (Contract No. 145/SPK-PF/RIK/08/2025).

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rissal Efendi	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Indrastanti Ratna Widiasari		✓		✓		✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest regarding the publication of this manuscript

DATA AVAILABILITY

The IoT-based rainfall and water level data supporting this study were obtained from sensor installations in Semarang, Indonesia. While the datasets are not openly accessible due to project restrictions, they are available from the corresponding author upon reasonable request.

REFERENCES

[1] W. Xu, J. Chen, X. J. Zhang, L. Xiong, and H. Chen, "A framework of integrating heterogeneous data sources for monthly streamflow prediction using a state-of-the-art deep learning model," *Journal of Hydrology*, vol. 614, p. 128599, Nov. 2022, doi: 10.1016/j.jhydrol.2022.128599.

[2] R. Parada and A. Sanz, "IoT-integrated deep learning for forecasting and decision support in reservoir water management under drought conditions," *Internet of Things*, vol. 34, p. 101780, Nov. 2025, doi: 10.1016/j.iot.2025.101780.

[3] M. Vadivel, R. V. Saraswathi, P. S. Lakshmi, R. R. Merugu, S. T., and S. Vivek, "Real-time coastal flood risk assessment using IoT-integrated satellite data and machine learning models for predicting flooding events and informing resilient coastal planning for Durban coastal region, South Africa," *Journal of African Earth Sciences*, vol. 233, p. 105856, Jan. 2026, doi: 10.1016/j.jafrearsci.2025.105856.

[4] B. Liu, Y. Li, M. Ma, and B. Mao, "A Comprehensive Review of Machine Learning Approaches for Flood Depth Estimation," *International Journal of Disaster Risk Science*, vol. 16, no. 3, pp. 433–445, Jun. 2025, doi: 10.1007/s13753-025-00639-0.

[5] L. B. L. Santos et al., "Machine learning-based hydrological models for flash floods: a systematic literature review," *Smart Construction and Sustainable Cities*, vol. 3, no. 1, Oct. 2025, doi: 10.1007/s44268-025-00071-9.

[6] A. Kumar, A. Bhatia, A. Kashyap, and M. Kumar, "LSTM Network: A Deep Learning Approach and Applications," *Advances in Social Networking and Online Communities. IGI Global*, pp. 130–150, Jun. 09, 2023. doi: 10.4018/978-1-6684-6909-5.ch007.

[7] O. Iparraguirre-Villanueva *et al*., "Text prediction recurrent neural networks using long short-term memory-dropout," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS),* vol. 29, no. 3, p. 1758, Mar. 2023, doi: 10.11591/ijeecs.v29.i3.pp1758-1768.

[8] I. M. Hayder *et al.,* "An Intelligent Early Flood Forecasting and Prediction Leveraging Machine and Deep Learning Algorithms with Advanced Alert System," *Processes*, vol. 11, no. 2, p. 481, Feb. 2023, doi: 10.3390/pr11020481.

[9] F. Granata and F. Di Nunno, "Neuroforecasting of daily streamflows in the UK for short- and medium-term horizons: A novel insight," *Journal of Hydrology*, vol. 624, p. 129888, Sep. 2023, doi: 10.1016/j.jhydrol.2023.129888.

[10] J. Li and X. Yuan, "Daily Streamflow Forecasts Based on Cascade Long Short-Term Memory (LSTM) Model over the Yangtze River Basin," *Water*, vol. 15, no. 6, p. 1019, Mar. 2023, doi: 10.3390/w15061019.

[11] Y. Zou, J. Wang, P. Lei, and Y. Li, "A novel multi-step ahead forecasting model for flood based on time residual LSTM," *Journal of Hydrology*, vol. 620, p. 129521, May 2023, doi: 10.1016/j.jhydrol.2023.129521.

[12] H. Huang *et al.,* "Improving the explainability of CNN-LSTM-based flood prediction with integrating SHAP technique," *Ecological Informatics*, vol. 84, p. 102904, Dec. 2024, doi: 10.1016/j.ecoinf.2024.102904.

[13] J. Li, G. Wu, Y. Zhang, and W. Shi, "Optimizing flood predictions by integrating LSTM and physical-based models with mixed historical and simulated data," *Heliyon*, vol. 10, no. 13, p. e33669, Jul. 2024, doi: 10.1016/j.heliyon.2024.e33669.

[14] Y. Zhang, Z. Gu, J. V. G. Thé, S. X. Yang, and B. Gharabaghi, "The Discharge Forecasting of Multiple Monitoring Station for Humber River by Hybrid LSTM Models," *Water*, vol. 14, no. 11, p. 1794, Jun. 2022, doi: 10.3390/w14111794.

[15] I. Ebtehaj and H. Bonakdari, "CNN vs. LSTM: A Comparative Study of Hourly Precipitation Intensity Prediction as a Key Factor in Flood Forecasting Frameworks," *Atmosphere,* vol. 15, no. 9, p. 1082, Sep. 2024, doi: 10.3390/atmos15091082.

[16] I. R. Widiasari and R. Efendi, "Utilizing LSTM-GRU for IOT-Based Water Level Prediction Using Multi-Variable Rainfall Time Series Data," *Informatics*, vol. 11, no. 4, p. 73, Oct. 2024, doi: 10.3390/informatics11040073.

[17] M. Halim, M. Wook, N. Hasbullah, N. Razali, and H. Hamid, "Comparative Assessment of Data Mining Techniques for Flash Flood Prediction," *International Journal of Advances in Soft Computing and Applications*, vol. 14, no. 1, pp. 126–145, Apr. 2022, doi: 10.15849/IJASCA.220328.09.

[18] A. Zabardast, E. G. Tamer, Y. A. Son, and A. Yılmaz, "An automated framework for evaluation of deep learning models for splice site predictions," *Scientific Reports*, vol. 13, no. 1, p. 10221, Jun. 2023, doi: 10.1038/s41598-023-34795-4.

[19] C. S. Alfredo and D. A. Adytia, "Time Series Forecasting of Significant Wave Height using GRU, CNN-GRU, and LSTM," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 776–781, Oct. 2022, doi: 10.29207/resti.v6i5.4160.

[20] A. Gasparin, S. Lukovic, and C. Alippi, "Deep learning for time series forecasting: The electric load case," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 1, pp. 1–25, Sep. 2021, doi: 10.1049/cit2.12060.

[21] A. Yildirim, M. Bilgili, and A. Ozbek, "One-hour-ahead solar radiation forecasting by MLP, LSTM, and ANFIS approaches," *Meteorology and Atmospheric Physics*, vol. 135, no. 1, Dec. 2022, doi: 10.1007/s00703-022-00946-x.

[22] N. Li, H. Sheng, P. Wang, Y. Jia, Z. Yang and Z. Jin, "Modeling Categorized Truck Arrivals at Ports: Big Data for Traffic Prediction," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 2772-2788, March 2023, doi: 10.1109/TITS.2022.3219882.

[23] V. Zelios, P. Mastorocostas, G. Kandilogiannakis, A. Kesidis, P. Tselenti, and A. Voulodimos, "Short-Term Electric Load Forecasting Using Deep Learning: A Case Study in Greece with RNN, LSTM, and GRU Networks," *Electronics*, vol. 14, no. 14, p. 2820, Jul. 2025, doi: 10.3390/electronics14142820.

[24] J. Zang *et al.,* "Prediction Model of Carbon Dioxide Concentration in Pig House Based on Deep Learning," *Atmosphere*, vol. 13, no. 7, p. 1130, Jul. 2022, doi: 10.3390/atmos13071130.

[25] T. Akilan and K. M. Baalamurugan, "Automated weather forecasting and field monitoring using GRU-CNN model along with IoT to support precision agriculture," *Expert Systems with Applications*, vol. 249, p. 123468, Sep. 2024, doi: 10.1016/j.eswa.2024.123468.

[26] U. S. Kumar *et al.,* "Fusion of MobileNet and GRU: Enhancing Remote Sensing Applications for Sustainable Agriculture and Food Security," *Remote Sensing in Earth Systems Sciences*, vol. 8, no. 1, pp. 118–131, Dec. 2024, doi: 10.1007/s41976-024-00183-3.

[27] T. Liang, M. Chen, D. Mi, and Y. Jing, "A novel GRU-Informer hybrid model with high and low frequency for multi-wind farm wind speed prediction," *International Journal of Green Energy*, vol. 22, no. 16, pp. 3762–3782, Jul. 2025, doi: 10.1080/15435075.2025.2533524.

[28] B. Mao, S. Tao, and B. Li, "Grain Temperature Prediction Based on GRU Deep Fusion Model," *International Journal of Information Technology & Decision Making*, vol. 24, no. 03, pp. 797–815, Jan. 2024, doi: 10.1142/s0219622023410031.

[29] B. Liu *et al.,* "A Hybrid VMD-BO-GRU Method for Landslide Displacement Prediction in the High-Mountain Canyon Area of China," *Remote Sensing*, vol. 17, no. 11, p. 1953, Jun. 2025, doi: 10.3390/rs17111953.

[30] Y. Zhao *et al.,* "Spatio-temporal prediction of groundwater vulnerability based on CNN-LSTM model with self-attention mechanism: A case study in Hetao Plain, northern China," *Journal of Environmental Sciences*, vol. 153, pp. 128–142, Jul. 2025, doi: 10.1016/j.jes.2024.03.052.

[31] H. Malik, J. Feng, P. Shao, and Z. A. Abduljabbar, "Improving flood forecasting using time-distributed CNN-LSTM model: a time-distributed spatiotemporal method," *Earth Science Informatics*, vol. 17, no. 4, pp. 3455–3474, Aug. 2024, doi: 10.1007/s12145-024-01354-y.
[32] S. Kumar, V. Kour, A. Verma, R. K. Sah and R. Misra, "Enhanced Forecasting Accuracy: Leveraging GRU+BiLSTM with Attention for Precise Floodwater Level Prediction," *2024 5th International Conference on Image Processing and Capsule Networks (ICIPCN)*, Dhulikhel, Nepal, 2024, pp. 108-115, doi: 10.1109/ICIPCN63822.2024.00026.
[33] Y. Shao, J. Chen, T. Zhang, T. Yu, and S. Chu, "Advancing rapid urban flood prediction: a spatiotemporal deep learning approach with uneven rainfall and attention mechanism," *Journal of Hydroinformatics*, vol. 26, no. 6, pp. 1409–1424, May 2024, doi: 10.2166/hydro.2024.024.
[34] T. T. Le, T. Q. Vo, and J. Kim, "An Attention-Enhanced Bivariate AI Model for Joint Prediction of Urban Flood Susceptibility and Inundation Depth," *Mathematics*, vol. 13, no. 16, p. 2617, Aug. 2025, doi: 10.3390/math13162617.
[35] M. Zhang, Z. Zhang, X. Wang, Z. Liao, and L. Wang, "The Use of Attention-Enhanced CNN-LSTM Models for Multi-Indicator and Time-Series Predictions of Surface Water Quality," *Water Resources Management*, vol. 38, no. 15, pp. 6103–6119, Dec. 2024, doi: 10.1007/s11269-024-03946-1.
[36] F. Wang, W. Wang, W. Bi, W. Lin, and D. Zhang, "Dual-Stage Attention-Based LSTM Network for Multiple Time Steps Flood Forecasting," *Proceedings of IAHS*, vol. 386, pp. 141–146, Apr. 2024, doi: 10.5194/piahs-386-141-2024.
[37] A. S. Atmaja, N. F. Muzakki, and Z. D. Oktavian, "Water Level Forecasting Using Long Short-Term Memory with Multi-Head Attention Mechanism," *Seminar Nasional Official Statistics*, vol. 2024, no. 1, pp. 625–636, Nov. 2024, doi: 10.34123/semnasoffstat.v2024i1.2125.
[38] Z. Vizi *et al*., "Water level prediction using long short-term memory neural network model for a lowland river: a case study on the Tisza River, Central Europe," *Environmental Sciences Europe*, vol. 35, no. 1, Nov. 2023, doi: 10.1186/s12302-023-00796-3.
[39] M. Cho, C. Kim, K. Jung, and H. Jung, "Water Level Prediction Model Applying a Long Short-Term Memory (LSTM)–Gated Recurrent Unit (GRU) Method for Flood Prediction," *Water,* vol. 14, no. 14, p. 2221, Jul. 2022, doi: 10.3390/w14142221.
[40] A. Dehghani *et al.*, "Comparative evaluation of LSTM, CNN, and ConvLSTM for hourly short-term streamflow forecasting using deep learning approaches," *Ecological Informatics*, vol. 75, p. 102119, Jul. 2023, doi: 10.1016/j.ecoinf.2023.102119.

BIOGRAPHIES OF AUTHORS

Rissal Efendi earned his Master of Information System from Diponegoro University. Since 2018, he has been a lecturer in Pedagogy on Informatics Engineering and Computer Science Study Program, Satya Wacana Christian University. He has working experiences as a computer network engineer in some private companies. His research interests are in the field of computer networks, cyber security, machine learning, deep learning and pervasive computing. Now he focusses on conducting IoT-based flood disaster prediction research in Semarang, Indonesia. He can be contacted at email: rissal.efendi@uksw.edu.

Indrastanti Ratna Widiasari was born in Sukoharjo, Central Java, Indonesia in 1978. She earned her Doctoral Degree in Electrical Engineering study program Universitas Gadjah Mada Yogyakarta in 2018. She is a lecturer in Informatics Engineering Satya Wacana Christian University. Her research interests are in the field of computer networks, wireless sensor networks, smart technology, deep learning and pervasive computing. She can be contacted at email: indrastanti@uksw.edu.