SCIENTIFIC CULTURE
www.sci-cult.com
DOI: 10.5281/zenodo. 20660295

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 11814-11825
Open Access. Online & Print

FROM PREDICTION TO ACTION: A COMBINED LSTM-ATTENTION AND FUZZY DIJKSTRA APPROACH FOR ENHANCED FLOOD DISASTER MITIGATION

Muhammad Soleh[1*], Nur Ulfa Maulidevi[1], Ziadan Qowi1, Wiwid Widyanto[1],R Praditya Priyanto[1], Caecilia Cantica Gita Mileniae Ahen[1], Ilmiaty Khairunnisa[1], Della Ayu Adinanda[2], Rifnaldo Karangan[3], Nofriza Hastuti[3], Khusnul Khatimah[4]

1School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia.
Email: muhsoleh90@gmail.com, maulidevi@itb.ac.id, ziadan.qowi97@gmail.com, wiwid.school@gmail.com, adenpraditya@gmail.com, caeciliaahen@gmail.com, ilmiatykhairunnisa@gmail.com
2Faculty of Civil and Environmental Engineering, Institut Teknologi Bandung, Bandung, Indonesia.
Email: dellaadinanda@gmail.com
3Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung, Bandung, Indonesia.
Email: rifnaldokarangan@gmail.com, nofrizahastuti@gmail.com
4Faculty of Earth Sciences and Technology, Institut Teknologi Bandung, Bandung, Indonesia.
Email: text.khusnul@gmail.com

Received: 04/04/2026
Accepted: 20/05/2026

Corresponding author: Muhammad Soleh
Email: ID: muhsoleh90@gmail.com

ABSTRACT

Urban flooding prevention in cities such as Jakarta necessitates efficient early warning and response logistics, which conventional approaches tend to divide into time-based forecasting and space-oriented evacuation routing. This paper presents a comprehensive disaster management system that integrates time-based water-level forecasting with risk-aware evacuation routing. This paper proposes using a Long Short-Term Memory (LSTM) neural network with an Attention mechanism to identify temporal dependencies in hydrometeorological time series. Predicted water levels serve as inputs for a Fuzzy Dijkstra's algorithm. Road sections in Jakarta's urban graph, comprising 94,038 nodes and 216,054 edges, are represented as Trapezoidal Fuzzy Numbers to apply time-based risk penalties to flooded routes based on predicted water levels. Experimental results indicate that the Attention mechanism effectively addresses overfitting issues and lowers forecasting errors. In addition, the forecasting structure with optimal performance comprises three layers and 128 hidden units and has attained the lowest error rate of 0.85%, an MAE of 0.0112, and an RMSE of 0.0340. For instance, in an overflow scenario where the estimated distance from the starting point to the goal was 0.6 meters, the classical Dijkstra algorithm was unable to find a feasible route because impassable areas directly crossed the flooded areas. On the other hand, the Fuzzy Dijkstra algorithm identified safe routes by avoiding high-risk flooded zones, even though the cost was relatively higher at 889.36 seconds

KEYWORDS: *: Flood Prediction, LSTM-Attention, Fuzzy Dijkstra, Evacuation Routing, Disaster Mitigation*

INTRODUCTION

Indonesia is a tropical state characterized by unique hydrometeorological conditions; thus, the annual cycle features dry and wet seasons. In this way, the geographic features create serious challenges for the occurrence of hydrological events, including floods.

The overflow of rivers has substantial environmental effects, including economic and financial problems and threats to human life. With increasingly dynamic environments, conventional methods prove insufficient, failing to meet modern standards for predicting and managing hydrometeorological phenomena. Therefore, there is a pressing need to use smart technology to produce reliable forecasts.

Flood data is usually represented as a time series that combines water-level and meteorological parameters. Cyclicality and a high degree of similarity characterize these datasets. Nowadays, there is a tendency to apply LSTM neural networks, as they have unique gate control capabilities that allow managing long-term dependencies in time series.

Many authors prove the effectiveness of recurrent models for this purpose. For example, according to Widiasari and Efendi (2024), a hybrid LSTM-GRU network achieves high performance, with an accuracy of 92.5% and an R^2 of 0.93–0.94. Zhang et al. (2023) report the computational speed of GRU models when predicting runoff, maintaining the same R^2 value of 0.7232 for the next 24-hour forecast.

In addition, new techniques, such as the Transformer architecture, have been used in this domain. According to Xu et al. (2026), TL-Transformer models are especially effective in scenarios with limited training data; the authors report NSE values above 0.75 with this approach. However, despite their superior performance compared to regular LSTMs in detecting high-dimensional relations, Transformer models incur a high computational cost and require much more memory.

On the other hand, LSTMs remain a computationally inexpensive and stable solution for forecasting floods without relying on excessive computation or transfer learning. Although the LSTM method effectively captures long-term dependencies, it also has limitations that prevent it from selecting the most relevant components of the information in the sequence of data. In this regard, the usage of the Attention Mechanism is considered one of the promising ways to improve the situation.

Despite the fact that there are only a few examples of the application of the Attention Mechanism together with LSTM to direct flood prediction problems, studies across different spheres demonstrate the high efficiency of this technology combination.

For example, according to the results of Kang et al. (2023) on developing a predictive model for the attitude and position of a shield machine in tunneling, integrating an LSTM and an Attention Mechanism improved the R-square value from 0.625 to 0.736. It decreased the RMSE from 3.31 to 2.24. The mentioned improvements can be assessed in terms of a 20% increase in accuracy and a 32% decrease in error rate, demonstrating the ability of the attention mechanism to select the most important input parameters.

Besides predicting potential disasters, the proposed study offers practical guidelines for responding to disasters by identifying appropriate evacuation points. Upon predicting a potential flood situation, the system suggests appropriate evacuation points using the Fuzzy Dijkstra Algorithm.

The Fuzzy Dijkstra algorithm plays an important role in addressing uncertainties in route selection during emergencies. Unlike the conventional Dijkstra algorithm, which considers only the shortest route, the Fuzzy Dijkstra algorithm enables the system to account for dynamic variables, such as possible levels of inundation along a route.

The usefulness of this technique has been established in prior research. For example, Afandi et al. (2022) developed a fuzzy Dijkstra-based tsunami evacuation route model for vehicles in Bengkulu City by prioritizing variables such as distance and road width, while differentiating road lanes according to carrying capacity to optimize evacuation effectiveness.

The objective of this research is to examine the efficacy of the LSTM algorithm with the attention mechanism in improving forecast accuracy for floods in Jakarta, compared with the standard LSTM algorithm alone. In addition, this research creates an evacuation suggestion system based on the Fuzzy Dijkstra Algorithm.

By incorporating both techniques, this research prioritizes safety rather than simply considering distance, since although the fuzzy logic inference system may require more processing time to calculate the route, the resulting routes ensure evacuation points are away from flooded areas. The novelty of this research lies in its end-to-end integration, in which real-time floodwater level predictions are directly used as dynamic inputs to the Fuzzy Dijkstra algorithm. Unlike traditional static routing, this approach allows for adaptive evacuation pathfinding that accounts for predicted inundation risks, bridging the gap between hydrological forecasting and disaster response logistics.

RESEARCH METHOD

This research follows a systematic, iterative approach to achieve flood-mitigation objectives using the CRISP-DM methodology. Through an innovative combination of time prediction using LSTM with an Attention Mechanism and spatial optimization using Fuzzy Dijkstra, the approach progresses through phases of business and data understanding, data preparation, and modeling, culminating in evaluation to ensure precision and accuracy using metrics such as RMSE, MAE, and sMAPE before finally being deployed to a real-time application, as shown in Figure 1.

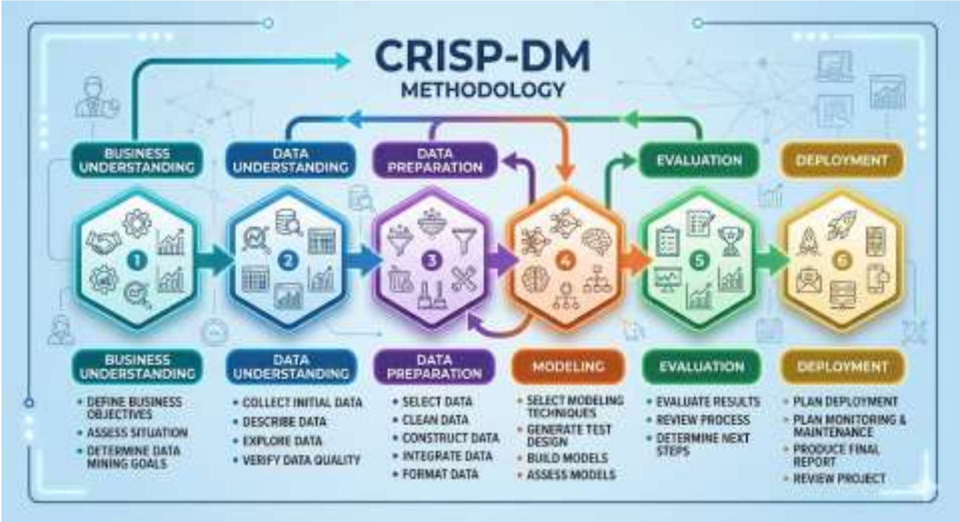


Figure 1: CRISP-DM.

The Business Understanding phase identifies the key problems associated with floods and converts them into technical solutions to mitigate the disasters. Therefore, understanding that inadequate warning time and inflexibility in evacuation routes are major issues.

This study offers the Regional Disaster Management Agency an integrated system with two objectives: to apply an LSTM model with an Attention Mechanism to make precise long-term water level predictions and to apply the Fuzzy Dijkstra algorithm to account for routing uncertainties. This study aims to make a tangible impact on flood mitigation by accounting for risks in Fuzzy Costs.

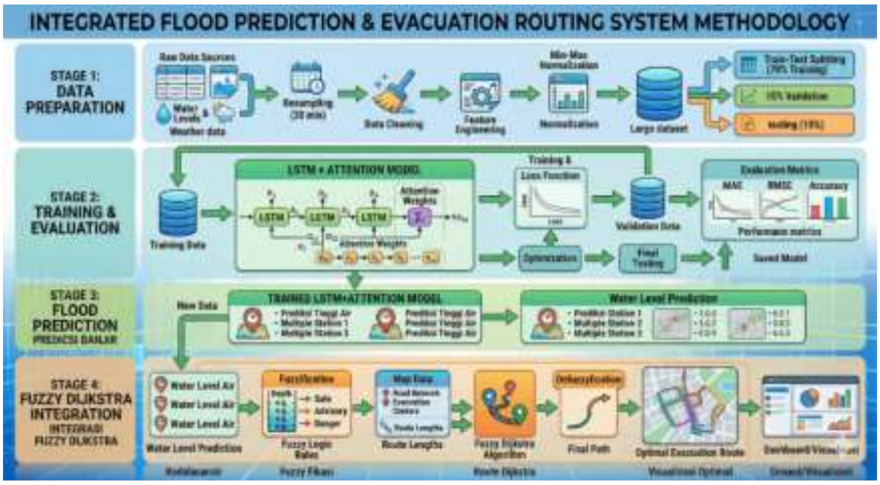


Figure 2: Integrated System Methodology Encompassing Four Main Stages: Data Preparation, LSTM-Attention Model Training, Flood Prediction, and Fuzzy Dijkstra-based Evacuation Routing.

The Data Understanding phase primarily focuses on collecting and validating the datasets required for developing the LSTM-Attention and Fuzzy Dijkstra models. The data integration in this research comprises two main domains: hydrometeorological data for temporal prediction and spatial data for optimization. The predictive data includes historical Water Level (WL) data from nine floodgates in Jakarta and is supplemented by meteorological data from Meteostat, including rainfall, temperature, humidity, and air pressure. The optimization data includes road network data collected via the OSMnx library (94,038 nodes and 216,054 edges) and coordinates for evacuation points.

The collected data shows characteristics which fit perfectly with the selected deep learning model and fuzzy logic. The water level data is non-stationary, with strong seasonal trends and a few extreme peaks during flooding. LSTM is selected as the highly suitable model for this data because the gating structure can effectively handle the data without the need to differ the data (Widiasari & Efendi, 2024). Exogenous variables such as rainfall and wind speed can also be included as inputs to the model to capture the complex behavior, with high lag effects, resulting in sudden increases in water level.

This quality assessment and outlier detection will ensure the data are of high quality before proceeding to the modeling stage. An initial data quality assessment in Python revealed no missing values across all key variables. Box plots also indicated the presence of extreme negative values in the dew point data and a highly skewed distribution in rainfall. However, these are key features in flood prediction, hence the need for Robust Scaling. The Data Preparation phase is concerned with data cleaning, normalization, and transformation in a manner best suited for LSTM-Attention and Fuzzy Dijkstra modeling. It commences with handling missing values in Water Level (WL) data caused by technical sensor failures during extreme events. These missing values are likely related to Missing Not at Random (MNAR) data, in which extreme water-level values may have caused sensor damage. Iterative Imputation methods can then be employed, as they leverage correlations between water levels and meteorological data to ensure flood risk is not underestimated.

Then, the feature scales are standardized using the Min-Max Scaling method to normalize variables with different scales, such as air pressure in the thousands and humidity in the tens, to the range [0, 1]. Time series transformation is also carried out using Resampling and Interpolation methods to convert irregular time intervals to a regular frequency of 30 minutes. In addition, Feature Engineering is performed using Lagged Features, which allow the LSTM model to effectively learn context through time-delay features.

The spatial aspect of GIS Data Integration is carried out by projecting the coordinates of the floodgates and evacuation sites onto the road network graph using the nearest nodes function. The initial weights for each road segment (edge) are established based on travel time, which is later integrated with Fuzzy Costs from the LSTM's prediction output. This ensures that each road segment has a dynamic resistance value, with segments predicted to be flooded assigned much higher weights to avoid them in the route optimization.

The Modeling phase integrates two main approaches to develop an integrated disaster mitigation system. It incorporates both temporal and spatial modeling. Temporal modeling is centered on the design of the Deep Learning architecture using LSTM with an Attention Mechanism to predict Water Levels. It is trained using water-level time-series data as the dependent variable and various exogenous hydrometeorological factors as independent variables to obtain predictions for future time periods.

At the same time, spatial modeling is performed by running the Fuzzy Dijkstra algorithm to optimize evacuation routes. The algorithm differs from other routing algorithms in that it incorporates fuzzy logic to handle uncertainties in route costs caused by flood inundation. The algorithm achieves this by transforming the predicted maximum inundation depth (Dmax) into time-based risk penalties and integrating them into the Fuzzy Cost values of every road in the network graph.

The novelty of this modeling process lies in the interaction between these two models, in which the LSTM model's water-level predictions are used as input to directly update the weights (costs) of the edges in the road network model. The road segments predicted to be flooded are automatically assigned a significantly higher risk weight using fuzzy membership functions, which forces the Dijkstra algorithm to guide users along drier roads. The output of this integrated modeling process is a dynamic model for recommending evacuation routes with minimized travel time and risk for the community.

The Evaluation phase evaluates the performance of the developed models. At the same time, it makes the required adjustments before implementing the system. In the case of the predictive model, performance is measured by evaluating RMSE, MAE, and sMAPE in time-series regression analysis. These values help evaluate the deviation between the predicted and actual water levels. Moreover, the Attention Mechanism is analyzed to evaluate the model's performance in distributing weights among variables.

Regarding route optimization, the evaluation compares the Fuzzy Dijkstra recommendations, which account for the predicted flood risk, with the conventional Dijkstra results, which account only for distance. The comparison ensures that the routes recommended are safe and realistic in emergencies. If the results are not optimal, the hyperparameters of the LSTM or the fuzzy logic membership functions are adjusted to improve performance.

The last phase of the methodology is the Deployment phase. In this phase, the research results are deployed to provide practical value for flood disaster mitigation. This phase involves developing an integrated application or prototype dashboard that integrates all the components of the research. The application is meant for real-time prediction results of Water Level (WL).

In addition to the prediction features, the system provides a visual representation of optimal evacuation route recommendations as a map. By leveraging the proper lead time generated by the prediction model and adaptive evacuation route recommendations, the system can improve preparedness and response to disasters. Further information regarding the prototype design, interface, and simulation case studies can be found in the results and discussion section.

RESULTS AND DISCUSSION

3.1 Results

3.1.1Experimental Setup

The experiments were conducted using a computer equipped with an Intel Core i3-10110U processor and 12 GB DDR4 RAM. The models were developed using Python 3.x along with TensorFlow, Keras, Pandas, NumPy, and OSMnx libraries. Before the modeling process, the dataset underwent preprocessing steps, including data normalization, handling missing values, and sequence generation for time-series forecasting. To improve model generalization and reduce overfitting, dropout regularization and early stopping were used during training. In addition, the Attention mechanism was integrated to enhance the model's ability to capture important temporal dependencies and spatial traffic patterns from the sequential data. The dataset was divided into training, validation, and testing subsets. Several hyperparameter configurations were evaluated, including variations in hidden units, dropout rates, sequence lengths, and Attention layer configurations. The Adam optimizer was utilized during training with Mean Squared Error as the loss function. Model performance was evaluated using RMSE, MAE, and sMAPE metrics.

Model Performance and Hyperparameter Stability

The dataset consists of water level measurements recorded at 40 monitoring stations. Descriptive analysis of the dataset showed heterogeneity in the water level scale, ranging from -6,554.0 cm to 6,288.0 cm. To address this issue, we used Min-Max normalization to scale all input variables to the [0, 1] range, ensuring stable gradient descent during LSTM training. The results of the correlation analysis showed that the linear associations between the weather characteristics (temperature, precipitation, humidity) and the water level values are extremely low, mostly near 0.00. Therefore, we used a non-linear LSTM architecture to capture hidden temporal dependencies.

Table 1: Sample Data Before Scaling.

floodgate id	timestamps	temp	dwpt	rhum	Prcp	wdir	wspd	pres	coco	water level
103	01/01/2022 0:00	26.6	23.7	84.0	0.0	265.0	5.5	1010.9	3.0	90.0
107	22/10/2022 14:10	25.7	23.7	89.0	1.6	145.0	3.6	1012.8	18.0	100.0
140	01/01/2022 0:00	26.8	23.5	82.0	0.0	260.0	11.2	1010.4	3.0	168.0
150	17/11/2022 14:26	27.0	23.3	80.0	0.0	268.0	7.4	1009.1	4.0	143.0
151	28/10/2022 18:26	19.6	15.9	79.0	0.0	0.0	0.0	1007.1	3.0	16.0
155	28/11/2023 11:20	25.3	24.3	94.0	0.1	158.0	4.7	1010.9	7.0	698.0
156	29/11/2023 18:52	26.6	25.0	91.0	0.3	0.0	0.0	1009.5	17.0	100.0
157	27/11/2023 8:08	27.7	23.7	79.0	0.0	335.0	10.4	1007.2	3.0	720.0
158	28/11/2023 13:13	25.0	24.3	96.0	1.2	248.0	8.3	1009.3	18.0	48.0

159	29/11/2023 11:48	25.3	24.4	95.0	0.8	188.0	10.8	1008.0	17.0	29.0

Table 2: Sample Data After Scaling.

floodgate id	timestamps	temp	dwpt	rhum	Prcp	wdir	wspd	pres	coco	water level
103	01/01/2022 00:00	0.339286	0.594937	0.722222	0.000000	0.736111	0.198556	0.601695	0.083333	0.152505
107	22/10/2022 14:10	0.351190	0.658228	0.851351	0.087912	0.402778	0.025899	0.760563	0.708333	0.544160
140	01/01/2022 00:00	0.300000	0.662338	0.753425	0.000000	0.722222	0.183007	0.580645	0.083333	0.264567
150	17/11/2022 14:26	0.263158	0.544304	0.648148	0.000000	0.743733	0.307054	0.449153	0.125000	0.275000
151	28/10/2022 18:26	0.214286	0.339506	0.666667	0.000000	0.000000	0.000000	0.256410	0.117647	0.032922
155	28/11/2023 11:20	0.584906	0.858025	0.892857	0.004630	0.438889	0.169675	0.503311	0.352941	0.949974
156	29/11/2023 18:52	0.333333	0.788889	0.854545	0.016484	0.000000	0.000000	0.513514	0.666667	0.237530
157	27/11/2023 08:08	0.470238	0.722222	0.726027	0.000000	0.930556	0.105799	0.366197	0.117647	0.861479
158	28/11/2023 13:13	0.329609	0.844720	0.948052	0.085106	0.688889	0.081452	0.456693	0.708333	0.573604
159	29/11/2023 11:48	0.333333	0.759615	0.924242	0.058394	0.522222	0.109868	0.378378	0.666667	0.151261

3.1.2 Flood Prediction Performance

First, model architectures and hidden layers. In contemporary times, the development of Beude Four different LSTM models with increasing depths were assessed to find the best model that could extract the most significant features from the temporal data: Model 1 (2 layers), Model 2 (3 layers), Model 3 (4 layers), and Model 4 (5 layers), each of which had the same number of hidden units, namely 64 per layer. These models were tested in their basic form and also with attention. The models using attention were created by adding a dedicated attention block comprising query and value transformations via Dense layers, additive attention to assign weights to each time point, and global average pooling to obtain the attention map. This increase in model depth led to a total rise in parameters from 41,305 in Model 1 to 148,761 in Model 4.

Second, comparative performance results. The integration of the Attention Mechanism consistently improved prediction accuracy and reduced error metrics across all models. The results are summarized in Table 3. The drastic decrease in RMSE, especially in Model 2 (from 0.0534 to 0.0342), clearly indicates that the attention mechanism is highly effective at mitigating high variance during prediction. This becomes very important for the flood early warning system, since the accuracy of prediction is paramount in this case (Kurniawan, 2024; PUPR, 2022; Srisulistiowati et al., 2021).

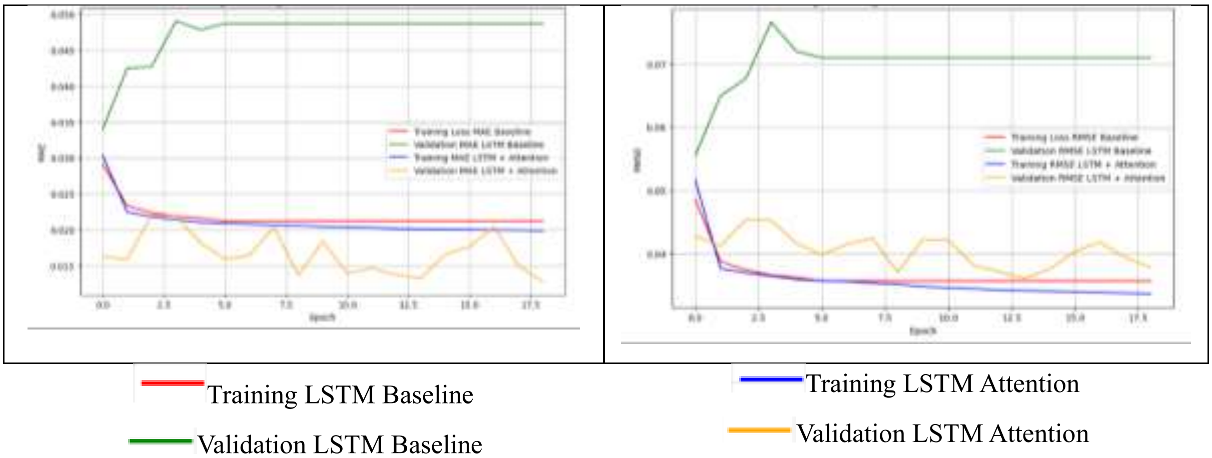
Figure 3A illustrates the convergence of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for Model 2, highlighting the superior generalization capabilities of the LSTM-Attention architecture compared to the baseline. While the Baseline LSTM (green line) exhibits a significant generalization gap, with validation errors surging early in training and reaching a high plateau of approximately 0.048 for MAE and 0.071 for RMSE, the integration of the Attention Mechanism effectively suppresses these errors. The LSTM-Attention model (blue and orange lines) maintains high stability, keeping the validation MAE consistently below 0.020 and the validation RMSE consistently below 0.045 throughout the 18-epoch process. These results demonstrate that the

Attention Mechanism successfully mitigates large prediction variances and prevents overfitting observed in standard models, ensuring robust, high-precision forecasting for flood early warning applications.

Figure 3B illustrates the comparison between the actual and predicted water levels at station 107. From Figure 3, it is evident that there is a notable difference in performance between the conventional and attention-based models. The Baseline LSTM graph on the left shows a noticeable discrepancy: the prediction line consistently fails to match the actual values, sometimes missing peaks and daily variations. On the other hand, the graph on the right, known as the LSTM-Attention graph, shows how closely the prediction line matches the actual values.

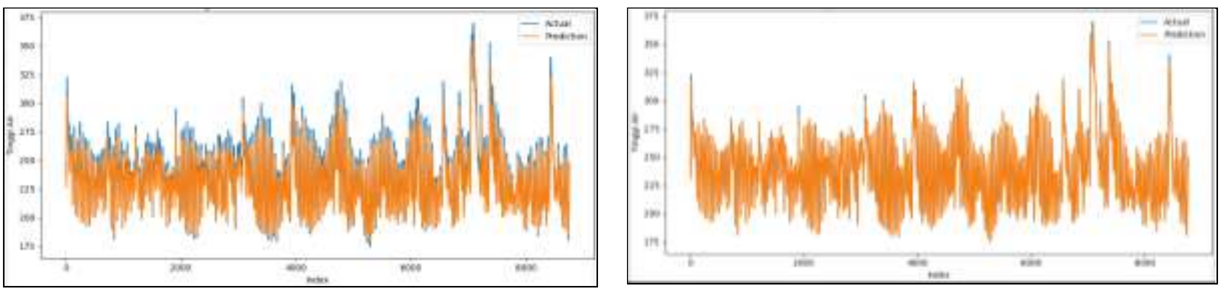
Table 3: Quantitative Summary of Model Performance and Prediction Accuracy.

Architecture	LSTM	MAE	RMSE	sMAPE (%)
Model 1 (2-Layer)	Baseline	0.0277	0.0447	26.9360
	Attention	0.0152	0.0371	17.8172
Model 2 (3-Layer)	Baseline	0.0332	0.0534	22.0454
	Attention	0.0129	0.0342	9.5919
Model 3 (4-Layer)	Baseline	0.0397	0.0505	30.0564
	Attention	0.0291	0.0464	20.9190
Model 4 (5-Layer)	Baseline	0.0330	0.0481	26.3613
	Attention	0.0203	0.0352	21.0384



A B

Figure 3: Performance Evaluation Of Model 2: (A) MAE Convergence During Training and Validation and (B) RMSE Convergence During Training and Validation.



A B

Figure 4: Comparison of Actual Versus Predicted Water Level Data for Model 2 at Station 107: (A) Baseline LSTM Results Showing Consistent Offsets and (B) LSTM-Attention Results Showing High Synchronization with Peak Values.

Third, hyperparameter optimization. Additional refinement was performed on Model 2 by optimizing the number of hidden units. It can be seen that adding more hidden units from 64 to 128 has resulted in significant improvements, with MAE of 0.0112, RMSE of 0.0340, and optimal sMAPE of 7.84%. This shows that adding more hidden units enables the model to understand better and identify complex temporal relationships in the time series.

Table 4: Hyperparameter Optimization Results for LSTM Hidden Units Comparing MAE, RMSE, and sMAPE Metrics.

Unit	MAE	RMSE	sMAPE (%)
64	0.0129	0.0342	9.5919
32	0.0126	0.0362	11.7349
128	0.0112	0.0340	7.8491
256	0.0140	0.0350	14.1201

Fourth, model evaluation and selection. Once the best architecture is found (Model 2 with a 3-layer LSTM, each layer with 128 hidden units), the findings can be applied to the Fuzzy Dijkstra Algorithm code. The process helps identify evacuation centers and create routes by assessing flooded routes to ensure the safest path is selected. By using accurate water levels as inputs to the fuzzy set, the algorithm can determine the impedance of each route in the road network, thereby avoiding any flood zones.

3.1.3Fuzzy Dijkstra Evacuation Routing

The evacuation routing experiment was conducted as a case study at the Cengkareng Drain water gate area. The simulation was designed to reflect critical flood conditions by integrating real-time and forecast water levels. The experiment established a maximum normal threshold of 190 cm (1.9 meters), while the actual scenario used a water level of 250 cm (2.5 meters), resulting in a predicted overflow depth of 0.6 meters (60 cm). This overflow value of 0.6 meters was utilized as the primary input for the Fuzzy Dijkstra algorithm.

In this model, road segment weights (W) are dynamically calculated based on fuzzy membership functions of the inundation depth, moving beyond the conventional Dijkstra approach that relies solely on physical distance. Specifically, the 0.6-meter overflow was processed through fuzzy logic to assign high impedance (time penalties) to affected road segments. This ensures that the evacuation recommendation system effectively identifies and bypasses hazardous paths, redirecting users to safer alternatives, even if those alternatives involve a longer distance.

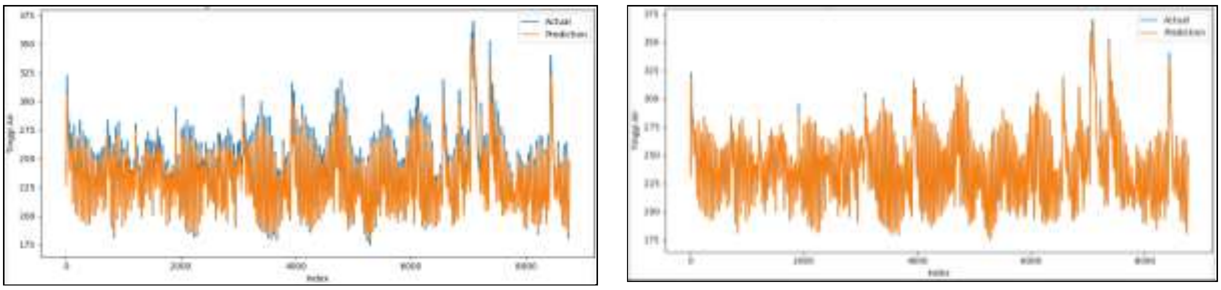
This section presents a comparative evaluation between the conventional Dijkstra algorithm and the proposed Fuzzy Dijkstra algorithm, utilizing a real-world evacuation scenario starting from the P.A. Cengkareng Drain area. The experiment was conducted by simulating a critical flood condition where a deviation between the actual water level (250 cm) and the maximum normal threshold (190 cm) resulted in a predicted overflow of 0.6 meters. This overflow value served as the primary dynamic input for the fuzzy-logic system. The results of the two routing approaches are summarized in Table 5.

Table 5: Comparison Summary of Evacuation Routing Experiments.

Comparison Parameter	Conventional Dijkstra	Proposed Fuzzy Dijkstra
Route Characteristics	Shortest Path (Static)	Safest Path (Adaptive/Fuzzy)
Number of Trajectory Nodes	32 Nodes	64 Nodes
Total Estimated Cost / Time	313.89 Seconds	889.36 Seconds (Fuzzy Cost)
Risk Consideration	Not Included	0.6 m Predicted overflow
Route Status	High Risk (Hazardous)	Safe (Verified)

Based on the routing results shown in Table 5, significant differences in navigation behavior were observed between the two algorithms. Computationally and geometrically, the conventional Dijkstra algorithm produces a more "efficient" route, with a shorter distance (32 nodes) and a faster processing time. This is because the conventional approach considers only static weights, in the form of physical distance between nodes. However, when these results are cross-referenced with the flood prediction model under the 0.6 m overflow scenario, a critical vulnerability is revealed. The "fastest" path recommended by the conventional Dijkstra algorithm intersects directly with low-elevation points predicted to be submerged. In a real-world evacuation scenario, this efficiency becomes ineffective because the route would be impassable, thereby neglecting the primary factor of human safety.

Conversely, the Fuzzy Dijkstra algorithm demonstrates a more adaptive and reliable behavior. Although the resulting route is longer and traverses more nodes, the algorithm successfully mitigates disaster risk by bypassing road segments with high inundation probability. The integration of fuzzy logic applies an "impedance penalty" to flooded paths based on the real-time inputs generated by the LSTM-Attention model (Levitin, 2012; Nguyen et al., 2021). As a result, the system automatically identifies alternative evacuation paths with higher elevations, reaching up to 8 meters above sea level, even if a longer detour is required. The total fuzzy cost of 889.36 seconds represents a rational trade-off to ensure that the recommended route remains free from water inundation. These findings demonstrate that the proposed Fuzzy Dijkstra model successfully transforms flood prediction data into actionable and safe navigation recommendations for flood-prone urban environments such as Jakarta.



A B

Figure 5: (A) Conventional Dijkstra Algorithm Result Showing the Shortest Path Selection and (B) Adaptive Routing Results Using Fuzzy Dijkstra to Mitigate Predicted Flood Risk Zones in Jakarta Road Network.

3.1.4 System Prototype Development

The system for flood monitoring and prediction is developed as a web-based application that combines the LSTM-Attention predictive model and the Fuzzy Dijkstra algorithm (Liu et al., 2019; Xu et al., 2026). The user interface is designed to provide comprehensive information for both the general public and relevant authorities. The application integrates real-time monitoring, flood prediction, evacuation mapping, and intelligent route recommendation into a unified platform to support disaster mitigation and emergency response.

The primary dashboard presents a real-time water-level monitoring table collected from multiple observation stations. The interface displays important information such as the location name, river source, and current water level. In addition, the dashboard provides a "Next Prediction" feature that displays water-level forecasts for the next 30 minutes up to 2 hours. Each observation station is equipped with an automated status indicator and direct access to evacuation route information, enabling users to assess flood conditions and respond quickly and appropriately.

To ensure consistency in flood-status classification across different observation points, the system includes a parameter management module that defines upper and lower thresholds for each alert category. These parameters are customized for each location due to the differences in geographical characteristics and river discharge capacities. This adaptive threshold configuration enables the system to provide more accurate, location-specific flood warnings.

The application also includes an evacuation area mapping feature based on Geographic Information System (GIS) technology. Through this feature, users can view the locations of available evacuation shelters directly on an interactive map. By selecting a shelter icon, users can access detailed information about the shelter name, the person in charge (PIC), and the contact number. This functionality improves accessibility to emergency facilities and supports coordinated evacuation efforts during flood events.

Another important component of the system is the evacuation route recommendation feature. Using the user's current position and the selected evacuation destination, the system applies the Fuzzy Dijkstra algorithm to determine the safest evacuation route. The generated route is displayed directly on the map, along with an estimated travel time, enabling users to evacuate efficiently while avoiding areas predicted to be inundated. The integration of fuzzy logic allows the system to dynamically adapt routing decisions based on predicted flood conditions, thereby improving evacuation safety during emergencies.

In addition, the system provides a graphical visualization of historical and predicted water-level data. The visualization shows the actual water level as a solid line and the predicted water level as a dashed line. The graph is also complemented with flood alert indicators such as “*waspada*,” “*siaga*,” and “*bahaya*.” Through this visualization, users can easily observe water-level trends and anticipate potential flooding conditions, enabling earlier preparation and decision-making before the disaster reaches critical levels.

Figure 6: (A) Home Interface and Real-Time Water Level Monitoring Dashboard, (B) Water Level Threshold Parameters per Observation Station, (C) Distribution of Evacuation Shelters on the Digital Map, (D) Visualization of Recommended Evacuation Routes using the Fuzzy Dijkstra Algorithm, and (E) Graphical Comparison of Actual vs. Predicted Water Level Data.

3.2.1Discussion

The experiment’s findings show that the configuration of hidden layers has a critical impact on the generalization ability of the proposed neural network. In particular, it is noted that 128 is the best threshold in terms of performance when used in an LSTM-Attention model. However, using fewer hidden layers (i.e., 32 and

64) has proved ineffective at capturing the complex temporal relations in Jakarta's hydrological data. In contrast, a higher number of hidden layers (i.e., 256) negatively affected the overall performance, with the error rate rising from 0.85% to 1.85%.

The inclusion of the Attention Mechanism offers a significant benefit when dealing with the heterogeneity of sensor data. According to the statistical study in the thesis, there is extremely high variance in some locations, such as Pintu Air 166, which has a standard deviation of 514.31 cm. Conversely, there is an extremely stable location, such as Pintu Air 169, with a standard deviation of 11.43 cm. The Attention Mechanism resolves this problem by giving more weight to important time instances and high-variance sensors in a surging event.

The dilemma between travel time and safety is an important aspect in responding to any disaster. The classical Dijkstra algorithm determines the optimal route having a total duration of 313.89 seconds; however, it does not consider the traversability of roads based on the presence of flood. On the other hand, the modified Fuzzy Dijkstra algorithm, which incorporates a penalty period of 5 minutes per risk score in the a4 trapezoidal fuzzy variable, takes a total route duration of 889.36 seconds (Deng et al., 2012). Despite taking a longer time route, the algorithm ensures that evacuees are routed in such a way that they bypass the "Extreme" risk area (Dmax > 0.50 m).

This system, as described herein, can turn water-level forecasts into intelligent solutions through its end-to-end process. By predicting water levels for fuzzy-logic routing, a proactive solution enables emergency agencies to take preventive measures before any danger arises. The road network analysis, conducted using over 94,038 nodes, shows how the algorithm can handle a large-scale city graph under varying flood-risk conditions. This enables the generation of evacuation recommendations as the flood moves from upstream to downstream areas.

CONCLUSION

The integration of the Attention Mechanism with the LSTM architecture, followed by the application of the Fuzzy Dijkstra algorithm, provides a robust solution for urban flood mitigation. Based on the comprehensive evaluation and analysis, it can be concluded that the LSTM-Attention model achieves its highest accuracy when configured with 128 hidden units, yielding a minimal error rate of 0.85%. This configuration effectively captures the non-linear temporal dependencies in Jakarta's water level data without overfitting. Furthermore, the Attention Mechanism significantly enhances the system's capability to process heterogeneous sensor data. By dynamically assigning greater weights to critical temporal changes, the model successfully manages extreme fluctuations, such as the 514.31 cm standard deviation observed at high-variance stations. This capability ensures more reliable early warning predictions during rapid flood surge events. The proposed Fuzzy Dijkstra algorithm also successfully bridges the gap between hydrological prediction and emergency response systems. By incorporating a risk-based time penalty using the a4 parameter in the fuzzy trapezoidal number, the system prioritizes evacuee safety during route calculation. Although the generated evacuation route has a longer travel time of 889.36 seconds than the shortest conventional route of 313.89 seconds, the proposed method effectively avoids "Extreme" risk zones where the predicted inundation depth exceeds 0.50 meters (Dmax > 0.50 m). In addition, the system demonstrates strong computational efficiency and scalability in processing large-scale urban transportation networks consisting of more than 94,000 nodes across Jakarta. This integration provides disaster management authorities with a proactive decision-support system that dynamically adapts to real-time flood progression and changing environmental conditions.

ACKNOWLEDGEMENTS

The lead author acknowledges the financial support provided by the Indonesia Endowment Fund for Education (*Lembaga Pengelola Dana Pendidikan* - LPDP) during the Master's research upon which this article is based.

REFERENCES

1. Afandi, N., Yosmar, S., & Mayasari, Z. M. (2022). Penentuan jalur evakuasi tsunami bagi pengendara di kota Bengkulu menggunakan algoritma fuzzy Dijkstra. Teorema: Teori Dan Riset Matematika, 7(1), 139. https://doi.org/10.25157/teorema.v7i1.6786

2. Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. Applied Soft Computing Journal, 12(3), 1231–1237. https://doi.org/10.1016/j.asoc.2011.11.011

3. Kang, Q., Chen, E. J., Li, Z. C., Luo, H. Bin, & Liu, Y. (2023). Attention-based LSTM predictive model for the attitude and position of shield machine in tunneling. Underground Space (China), 13, 335–350. https://doi.org/10.1016/j.undsp.2023.05.006

4. Kurniawan, A. (2024). Perbandingan analisis debit banjir dengan menggunakan aplikasi hec-ras dan EPA SWMM. Cakrawala Repositori IMWI, 7(2), 567–577. https://doi.org/10.52851/cakrawala.v7i02.641

5. Levitin, A. (2012). Introduction to the design and analysis of algorithms. Addison-Wesley.

6. Liu, Z., Zhou, W., & Li, H. (2019). AB-LSTM: Attention-based bidirectional lstm model for scene text detection. ACM Transactions on Multimedia Computing, Communications and Applications, 15(4). 107. https://doi.org/10.1145/3356728

7. PUPR, K. (2022). Modul 7 pengelolaan resiko banjir, pelatihan perencanaan teknis sungai. Pusat Pengembangan Kompetensi Sumber Daya Air dan Permukiman.

8. Srisulistiowati, D. B., Suryani, R., & Rejeki, S. (2021). Peningkatan pendapatan bunga bersih dari pengaruh aktiva produktif pada bank Central Asia tahun 2017-2021. Jurnal Kajian Ilmiah, 21(4), 455–466.

9. Widiasari, I. R., & Efendi, R. (2024). Utilizing LSTM-GRU for IoT-Based water level prediction using multi-variable rainfall time series data. Informatics, 11(4), 73. https://doi.org/10.3390/informatics11040073

10. Xu, S., Wang, H., Xu, H., Wu, Z., Zhang, X., Zhou, Y., & Xue, W. (2026). Urban flood prediction model based on explainable deep learning and attention mechanism. International Journal of Disaster Risk Science, 17, 161–181. https://doi.org/10.1007/s13753-026-00691-4

11. Zhang, Y., Zhou, Z., Van Griensven Thé, J., Yang, S. X., & Gharabaghi, B. (2023). Flood forecasting using hybrid LSTM and GRU models with lag time preprocessing. Water (Switzerland), 15(22), 3982. https://doi.org/10.3390/w15223982..

1