Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3911~3919
e-ISSN: 2548-6861

Performance Comparison of Random Forest, Support Vector Machine, and K-Nearest Neighbors Algorithms in Ten-Minute Rainfall Prediction for Urban Flood Mitigation in South Tangerang

Tri Nurmayati
Informatics Engineering, Pamulang University
trinurmayati@gmail.com

Info Article

Article history:

Accepted 2025-08-15
Revised 2026-07-25
Accepted 2026-08-12

Keywords:

KNN,
Orange,
Rainfall,
Random Forest,
SVM.

ABSTRACT

Rainfall prediction with very high temporal resolution, such as ten-minute intervals, is of great urgency in urban flood mitigation, especially in densely populated areas such as South Tangerang. Most previous studies still focus on daily or monthly rainfall prediction, so this study attempts to fill this gap by comparing the performance of three machine learning algorithms: Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) in predicting ten-minute rainfall. A dataset of 52,703 entries was obtained from an automated weather station (AWS) with predictor variables of temperature, humidity, and air pressure. The analysis process was carried out through preprocessing stages (missing value imputation, Min–Max normalization), data splitting (80:20), and 10-fold cross-validation using Orange Data Mining software. Model performance evaluation used Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2). The results showed that Random Forest performed best with an RMSE of 0.671 and MAE of 0.090, followed by KNN with an RMSE of 0.683 and MAE of 0.083, while SVM performed significantly lower (RMSE of 2.553 and MAE of 2.511). However, the negative R^2 values for all models indicate limitations in explaining rainfall data variability, which is likely influenced by skewed data distribution (zero-inflated) and AWS sensor noise. These findings indicate that Random Forest is relatively more reliable than other models, but its accuracy is still not optimal for operational applications. Future research is recommended to adopt a hybrid or deep learning approach (e.g., LSTM), add other meteorological variables such as wind speed and solar radiation, and integrate the models into IoT-based early warning systems to support urban flood mitigation.

I. INTRODUCTION

A. Background

Rainfall is a key meteorological factor that significantly impacts environmental stability and human settlements. In tropical and rapidly urbanizing regions like Indonesia—specifically in densely populated satellite cities such as South Tangerang rainfall patterns tend to be highly localized and intense [1]. Due to extensive paved infrastructure and a high percentage of impervious surfaces, South Tangerang faces severe vulnerability to urban flash floods and waterlogging [2]. These hydrometeorological disasters frequently paralyze urban mobility, damage critical infrastructure, and inflict substantial economic losses on both municipal budgets and local commerce. Therefore, accurate rainfall prediction with very high temporal resolution is urgently required to support proactive urban disaster mitigation [2] [3] .

Predicting rainfall at a high temporal resolution (such as ten-minute intervals) presents extreme scientific complexity due to the highly localized micro-climate variability inherent to tropical urban areas [4]. Rapid urbanization, localized atmospheric heating over paved surfaces (the urban heat

island effect), and sudden convective cloud formation cause extreme spatial and temporal rainfall variability across South Tangerang. Traditional daily or monthly forecasting aggregations completely smooth out these critical micro-variations. Furthermore, in dense urban environments, the hydrological response time (time of concentration) is extremely short; thus, daily or even hourly forecasts are physically inadequate for capturing sudden cloudbursts that trigger immediate urban flooding.

This leaves a critical research gap in ultra-short temporal resolutions, which are essential for operational urban drainage management. Although recent studies between 2022 and 2024 have increasingly adopted Machine Learning (ML) for short-term precipitation forecasting [5] [6]the majority still rely on hourly radar or satellite imagery in temperate regions. To the authors' knowledge, this is the first study to investigate and compare supervised ML paradigms specifically for 10-minute resolution rainfall prediction in an Indonesian urban settlement using terrestrial Automatic Weather Station (AWS) data, providing a novel baseline for localized hydrometeorological early warning systems [7].

To bridge this gap, this study compares three foundational supervised learning algorithms chosen for their distinct computational paradigms: Random Forest (RF), Support Vector Machine (SVM/SVR), and K-Nearest Neighbours (KNN) [8]. RF represents an ensemble decision-tree approach robust against non-linear meteorological data and overfitting [9]; SVM is a margin-based learning model designed to identify optimal separation hyperplanes in complex feature spaces [10]; and KNN is a non-parametric, instance-based algorithm that predicts values based on local historical data similarities [11]. Comparing these diverse paradigms provides critical insight into which learning mechanism best adapts to high-resolution tropical weather data. To the authors' knowledge, this is the first study comparing these three ML paradigms for 10-minute resolution rainfall prediction in an Indonesian urban area using AWS data.

The South Tangerang region faces the challenges of urbanization and climate change, which increase its vulnerability to the impacts of extreme rainfall. Therefore, research focused on the development and comparison of high-resolution rainfall prediction models in this region is highly relevant and important [12]. The findings of this study are expected to not only contribute to the development of academic literature but also provide practical benefits for The Agency for Meteorology, Climatology, and Geophysics (BMKG) and the city government in supporting urban flood early warning systems.

B. Objectives

The explicit objectives of this study are as follows:

1) To implement and adapt three distinct machine learning algorithms—Random Forest (RF), Support Vector Machine (SVM/SVR), and K-Nearest Neighbors (KNN)—for ultra-short, ten-minute high-resolution rainfall prediction in South Tangerang.
2) To evaluate and systematically compare the predictive performance of these three models using standardized regression error metrics, specifically Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2).
3) To identify the most reliable algorithm capable of handling local micro-climate variability and zero-inflated terrestrial weather station data.
4) To formulate actionable, data-driven recommendations for municipal authorities and meteorological agencies (BMKG) in establishing an operational, IoT-integrated urban flood early warning architecture.

II. THEORETICAL FRAMEWORK

A. Rainfall

Rainfall is a crucial climate variable that impacts various sectors, including hydrology, agriculture, and disaster management [13]. According to the Meteorology, Climatology, and Geophysics Agency (BMKG), rainfall is the height of rainwater accumulated in a measuring instrument mounted on a flat, non-absorbent, impermeable, and runoff-free surface. In Indonesia, rainfall is measured in millimeters (mm). A rainfall measurement of 1 (one) millimeter means that in an area of one square meter on a flat surface, rainwater accumulates to a depth of one millimeter, equivalent to one liter of water. The amount of rainfall during a certain period is expressed in units such as mm/hour, mm/day, or mm/year.

The most common rainfall measuring instruments used by the BMKG are observatory rainfall gauges (standard ombrometers) and automatic rainfall gauges (Automatic Rain Gauges/ARGs or Automatic Weather Stations/AWSs). Additionally, there are semi-automatic instruments that record rainfall graphically on rainfall graph paper.

Figure 1. Rainfall Measuring Instrument

B. Conventional Rainfall Prediction Model

Traditionally, rainfall prediction has relied heavily on statistical models, such as multiple linear regression, and numerical weather prediction (NWP) models and time series analysis [4], which use historical relationships between variables to project future conditions. Although these models are simple and fast, they often struggle to capture the non-linear and complex nature of weather phenomena.

NWP models (e.g., WRF, GFS) [14] solve atmospheric physics equations to predict weather conditions. These

models are very comprehensive but require large computing power and detailed input data, making them less flexible for very short-term local forecasts without adequate infrastructure [15].

C. Machine Learning-Based Rainfall Prediction

With the advancement of computing, ML algorithms are increasingly being used for rainfall prediction, thanks to their ability to handle non-linear relationships and large data sets.

1. Random Forest (RF)

Random Forest is an ensemble method that builds multiple decision trees and combines their outputs for prediction [9]. This method is known for its resilience to overfitting and its ability to handle high-dimensional and non-linear data. Several studies have successfully implemented RF for rainfall prediction. For example, [16]used RF to predict daily rainfall in Bandung with satisfactory results. Another study [17] showed that RF outperformed other methods in short-term rainfall prediction in urban areas.

2. *Support Vector Machine (SVM)*

Support Vector Machine Support Vector Machine (SVM) is an effective supervised learning algorithm for classification and regression. Developed by Vapnik et al. in the 1990s [10], its principles were later adapted for regression as Support Vector Regression (SVR). In the context of data imputation, SVR serves as a predictive model that fills in missing values of a target variable based on learned relationships from other variables available in the dataset.

Unlike traditional linear regression that minimizes squared error, SVR aims to find the optimal function or hyperplane that best represents the relationship between input features (X) and target variables (Y), with an emphasis on a certain error tolerance (ϵ). The essence of SVR is to identify a hyperplane that not only minimizes errors outside the bounds (or margins) that are insensitive to ϵ, but also optimizes generalization by controlling model complexity [18] . This means that SVR penalizes errors outside the tolerance bounds but ignores errors within them, making SVR relatively robust to outliers within a certain bound.

In the context of rainfall prediction, SVR can be trained using available rainfall data as the target variable, with relevant meteorological parameters (e.g., temperature, humidity, atmospheric pressure from the same day, or rainfall data from a nearby station) as input features. SVR has been shown to be effective in rainfall prediction in various regions. For example, [19] showed that SVR can provide accurate rainfall predictions even with relatively small datasets. A study [20] comparing SVR with Artificial Neural Networks (ANN) for monthly rainfall prediction in Bogor found that SVR offers competitive performance.

3. K-Nearest Neighbors (KNN)

KNN is a nonparametric method that estimates values using the mean or mode of the k nearest neighbors in a feature space [21]. Despite its simplicity, KNN can be effective for both regression and classification problems. In the context of rainfall prediction, a study by Wijaya and Saputra (2022) used KNN for hourly rainfall forecasting and found that its performance was highly dependent on the choice of k and the distance metric. Another study [21] compared KNN with linear regression for daily rainfall prediction, showing that KNN can effectively capture local patterns.

D. Orange Software for Machine Learning

Orange is an open-source visual platform for data analysis, data mining, and machine learning. Developed by the University of Ljubljana, Slovenia, Orange is designed to facilitate data exploration, preprocessing, modeling, and visualization, even for users without a strong programming background [21]. Since its release, Orange has continued to evolve and has been recognized as a flexible and powerful tool in the data science community.

Orange operates on a widget-based architecture, where each widget represents a specific data mining or machine learning function (e.g., loading data, preprocessing, training a model, or visualizing results). Users create workflows by visually connecting widgets, forming a graphical representation of the data analysis process [21]. Its flexibility allows users to:

- Fast Data Exploration: Load datasets from various formats (.csv, Excel, databases) and instantly visualize distributions, descriptive statistics, and variable relationships through widgets such as data tables, distributions, or scatterplots.
- Data Preprocessing: Orange provides various widgets to handle crucial preprocessing steps, including missing value handling (e.g., Impute Missing Values with median imputation or KNN imputation), feature normalization or standardization (Normalize), outlier handling, and data transformation [21]. Efficient preprocessing ensures high-quality data before modeling.
- Machine Learning Implementation: Orange integrates various popular machine learning algorithms for classification, regression, clustering, and association. For regression and imputation purposes, Orange offers widgets such as linear regression, Support Vector Machine (configurable for SVR), and KNN. Users can easily adjust algorithm parameters through a graphical interface.
- Results Visualization: Orange offers advanced visualization widgets for analyzing and presenting model results, including line plots for time series, scatterplots for variable relationships, and various evaluation charts such as ROC curves or confusion matrices (for classification). These visualizations are invaluable for comparing model performance and understanding the characteristics of imputed data.
- Model Evaluation: The evaluation widget allows users to calculate Performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE)

for regression, and to perform cross-validation to obtain more robust performance estimates [21].

Over time, Orange has continuously updated and expanded its add-on ecosystem. Active community contributions have resulted in add-ons providing additional functionality for specialized domains such as Text Mining, Bioinformatics, or Time Series. This flexibility makes Orange relevant for a wide range of applications, from academic research to industrial data analysis [22]. Its ability to run custom Python scripts via the Python Script widget adds advanced capabilities for implementing algorithms previously unavailable natively in previous versions [21].

IV. RESEARCH METHOD

This study uses a quantitative comparative experimental design to assess and compare the performance of three machine learning algorithms: Random Forest (RF), Support Vector Regression (SVR), and K-Nearest Neighbors (KNN) in predicting ten-minute rainfall values in South Tangerang. All data processing and modeling were performed using Orange Data Mining, a visual software that enables machine learning implementation without programming.

A. Research Data

This dataset was obtained from an Automatic Weather Station (AWS) located in South Tangerang, covering the period from January to December 2024 with a 10-minute observation interval. The dataset contains 52,703 entries and includes the following variables:

- Rainfall (mm) - Target Variable
- Air temperature (°C) - Predictor variable
- Relative humidity (%) - Predictor variable
- Air pressure (mb) - Predictor variable

B. Research Workflow and Data Processing

To ensure data integrity and optimize algorithmic convergence, the raw AWS dataset underwent a rigorous preprocessing pipeline before model execution. The preprocessing methodology consisted of three critical stages:

1) Feature Selection and Target Definition: Explicit structural roles were assigned to the dataset attributes. Air temperature (˚C), relative humidity (%), and air pressure (mb) were designated as predictor features (X), while the ten-minute rainfall interval (mm) was defined as the target dependent variable (y). Station IDs and timestamps were preserved purely as structural metadata.
2) Missing Value Handling (Imputation): Data completeness is mandatory for supervised regression training. Initial auditing revealed that less than 0.5% of the 52,703 records contained missing values. Because the proportion of missingness was negligible, a listwise deletion (row filtering) method was applied to exclude incomplete instances. This ensured that the models were trained exclusively on empirically observed meteorological ground truth without introducing artificial imputation bias.
3) Feature Scaling (Normalization): Meteorological predictors inherently operate on vastly different numerical scales (e.g., pressure around 1000 mb vs. temperature around 30°C). Because distance-based algorithms like KNN and margin-based optimization in SVM are highly sensitive to feature magnitudes, Min-Max Normalization was applied to uniformly scale all predictor variables into a standardized range of [0,1]. This step prevented variables with larger absolute numbers (such as air pressure) from disproportionately dominating the Euclidean distance calculations or hyperplane positioning.

C. Data Splitting and Validation Strategy

To prevent model overfitting and evaluate generalization capabilities on unseen data, the preprocessed dataset was partitioned using an 80:20 random deterministic split:

- Training Sub-set (80%): Used exclusively for algorithmic learning, pattern extraction, and hyperparameter stabilization.
- Testing Sub-set (20%): Held out as unseen data to evaluate final operational prediction accuracy.

Furthermore, during the training phase, a 10-fold Cross-Validation ($k = 10$) resampling technique was integrated. The training data was divided into 10 equal subsets; each model was trained iteratively on 9 folds and validated on the remaining 1 fold. This multi-fold validation ensures that the reported error metrics (RMSE, MAE, and R^2) are statistically robust, stable across different data distributions, and not merely artifacts of a single random data split.

E. Model Implementation

The selection of these three algorithms is grounded in their complementary yet distinct machine learning paradigms, making them ideal for a comparative study on high-resolution hydrometeorological data [3] [23]:

- Random Forest (RF) — Ensemble Learning Approach: RF builds multiple decision trees and aggregates their outputs [17]. It was selected for its proven robustness in capturing highly non-linear relationships among meteorological variables (temperature, humidity, pressure) without making strict data distribution assumptions, while its bagging mechanism effectively prevents overfitting.
- Support Vector Regression (SVR) — Margin-Based Learning: Unlike standard regression that minimizes squared error, SVR constructs an optimal hyperplane within a multi-dimensional feature space using an error tolerance margin (ϵ) [18]. It was chosen to test whether margin-based optimization can effectively separate low-intensity rainfall patterns from ambient weather noise [18].
- K-Nearest Neighbors (KNN) — Instance-Based Learning: KNN is a non-parametric algorithm that relies

on feature-space proximity using Euclidean distance [24]. It was selected to evaluate whether micro-scale rainfall can be accurately predicted simply by recognizing localized, historical meteorological clustering patterns without building a complex global model.

F. Model Evaluation

The models were evaluated using three main performance metrics:

1. Root Mean Square Error (RMSE)

$$RSME = \sqrt{\frac{1}{N}\sum_{me=1}^{N} \left(And_{I\,am} - \widehat{And}_{I\,am}\right)^2}$$

Where:

- $And_{I\,am}$:Actual value of rainfall at the time of observation i
- $\widehat{And}_{I\,am}$: Predicted value at observation i
- n: Total number of observations
- $\left(And_{I\,am} - \widehat{And}_{I\,am}\right)^2$: Squared error for observation i

A smaller RMSE value indicates higher model accuracy. In rainfall forecasting, RMSE indicates how much the prediction deviates from the average actual value.

2. Mean Absolute Error (MAE)

$$IS = \frac{1}{N}\sum_{me=1}^{N} \left|And_{I\,am} - \widehat{And}_{I\,am}\right|$$

Where:

- $And_{I\,am}$: Actual value of rainfall at the time of observation i
- $\widehat{And}_{I\,am}$: Predicted value at observation i
- n: Total number of observations
- $\left|And_{I\,am} - \widehat{And}_{I\,am}\right|$: Absolute difference between actual value and predicted value

MAE reflects the average magnitude of the prediction error, regardless of its direction (positive or negative). Compared to RMSE, MAE is more tolerant of outliers. In the context of rainfall prediction, MAE helps assess the daily deviation of predictions from actual conditions. The smaller the MAE, the better the model's performance.

3. Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{\sum_{me=1}^{N} \left(And_{I\,am} - \widehat{And}_{I\,am}\right)^2}{\sum_{me=1}^{N} \left(And_{I\,am} - \underline{And}_{I\,am}\right)^2}$$

Where:

- $\underline{And}$: Average of all actual rainfall values
- Numerator: Total prediction error
- Denominator: Total variance in actual data

Interpretation of R^2:

- The R^2 value typically ranges from 0 to 1 for a model that fits the best regression line (although it can be negative in certain cases).
- $R^2 = 0$: Indicates that the regression model does not explain any variance in the target variable around its mean value. In other words, the model performs no better than simply predicting the mean value of the target variable.
- $R^2 = 1$: Indicates that the regression model explains all the variance in the target variable around its mean. This means that all data points lie exactly on the predicted regression line, a scenario that is very rare in real-world data.
- A higher R^2 value generally indicates a better performing model. For example, $R^2 = 0.75$ means that 75% of the total variance in the target variable can be explained by the regression model.

V. RESULTS AND DISCUSSION

To obtain accurate predictions and ensure their validity, data processing in Orange Data Mining is carried out in stages through several main components, each of which contributes significantly to forming the data structure, preparing the data to be suitable for learning by the model, and evaluating the performance of the model used.

The figure below illustrates the implementation and evaluation flow of an algorithm model using Orange Data Mining software. This workflow consists of variable selection, missing data imputation, normalization, data splitting, model training, performance evaluation using the Test & Score widget, and prediction.

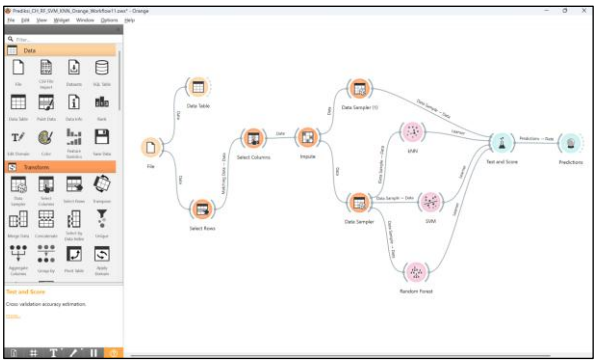

Figure 1. Model Processing and Evaluation Workflow in Orange

The following is a description of each stage of data processing along with a visual display of each step:

1. Files

Upload a dataset and assign data roles as metadata, features, and targets. This dataset consists of 52,703 rows, with rainfall as the target, while humidity, temperature, and

pressure serve as features. In this study, station IDs and timestamps are stored as metadata.

2. Select Row

The Select Rows widget is used to filter out incomplete data, specifically rows missing a target value (rainfall). This is important because supervised learning algorithms cannot train without a clearly defined target value. At this stage, the "rainfall has been determined" condition is implemented, meaning only rows with a defined target value will proceed to the next stage. This step helps maintain dataset quality and avoid errors during model training.

3. Select Column

At this stage, the Select Columns widget is used to define the input and output structures for the model training process. The temperature, humidity, and air pressure columns are selected as features because they are independent variables that can potentially influence rainfall values. Meanwhile, the rainfall column is designated as the target, the dependent variable to be predicted by the model. Column selection is done explicitly to ensure the algorithm only processes relevant attributes, thus avoiding confusion in the data structure. This stage serves as an important foundation before entering preprocessing and modeling.

4. Imputation

The Impute widget is used to handle missing values in input features such as temperature, humidity, and air pressure. During this process, Orange offers several auto-fill methods, such as using the mean value for numerical data. Imputation is crucial for preventing machine learning models from failing during training due to undefined values. In this study, only 0.5% of the data set contained missing data. Therefore, missing values were simply ignored or not filled in.

5. Data Sampling

At this stage, the Data Sampler widget splits the dataset into 80% training data and 20% testing data, using a random but reproducible split to ensure the evaluation results are replicable. This split is important for assessing the model's ability to generalize to previously unseen data.

For Data Sampler 1, cross-validation was applied, a method used to evaluate the performance of machine learning models, primarily to ensure that the model does not perform well only on a particular test set (thus avoiding overfitting and providing more stable performance estimates).

6. Test & Score

After completing preprocessing, normalization, and data splitting, three algorithms—Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)—were trained and evaluated using RMSE, MAE, and R^2, which are standard metrics for regression model accuracy.

Table 1. Evaluation Results Using RMSE, MAE, and R^2

Algorithm	RMSE	THE RE IS	R^2
K-*Nearest Neighbor*(k=5)	0,683	0,083	-0,102
SVM	2.553	2.511	-14.378
Random Forest	0,671	0,090	-0,062

The evaluation results of the three machine learning models—Random Forest (RF), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)—demonstrated significant differences in their predictive capabilities. Based on absolute error metrics, the Random Forest model performed best, achieving the lowest RMSE (0.671) and MAE (0.090). This indicates that the ensemble decision-tree approach effectively minimizes average prediction deviations. KNN achieved the second-best performance with an RMSE of 0.683 and an MAE of 0.083, reflecting a highly competitive accuracy level. In stark contrast, SVM performed significantly worse, generating an RMSE of 2.553 and an MAE of 2.511. This substantial margin of error confirms that standard margin-based regression struggling with hyper-parameter tuning is ill-suited for this specific dataset.

Although Random Forest and KNN successfully minimized absolute prediction errors, the most critical diagnostic indicator of model reliability in this study is the Coefficient of Determination (R^2). All three models produced consistently negative R^2 values: Random Forest (-0.062), KNN (-0.102), and SVM (-14.378). In formal regression analysis, a negative R^2 signifies that the residual sum of squares exceeds the total variance of the observed target variable. Practically, this proves that the trained machine learning models performed *worse* than a naive baseline model that simply predicts the constant historical mean of the rainfall data. Therefore, while low RMSE and MAE values suggest minimal everyday deviation, the negative R^2 values reveal that the models currently lack the mathematical reliability required to capture and explain the dynamic variance of 10-minute precipitation events independently.

The fundamental cause of the negative R^2 values is directly linked to the structural characteristics of high-resolution terrestrial weather data. First, 10-minute interval rainfall data is heavily zero-inflated; over 90% of the recorded observations represent non-raining conditions (0 mm). When supervised regression algorithms attempt to map continuous atmospheric predictors (temperature, humidity, and pressure) to a target variable heavily dominated by zeros, they fail to establish a distinct decision boundary between ambient "no-rain" states and sudden "low-intensity rainfall" events. Second, terrestrial Automatic Weather Station (AWS) sensors inevitably introduce environmental and technical noise—such as micro-fluctuations in pressure or humidity that occur independently of convective precipitation. Without broader

synoptic weather context, algorithms are easily misled by this noise. This vulnerability is most severe in SVM (R^2 of -14.378), where optimal separating hyperplanes are heavily over-penalized by outliers in sparse, noisy distributions.

Furthermore, the predictive capacity of the evaluated models is constrained by the limited dimensionality of the input feature space. The dataset relies solely on three surface-level predictors: air temperature, relative humidity, and air pressure. Fundamental thermodynamic and mechanical drivers of tropical convective rainfall—such as wind speed and direction, solar radiation, sunshine duration, and cloud cover—were unavailable in this dataset [25]. Without these critical atmospheric dynamics, the models lack the necessary physical context to anticipate sudden localized cloudbursts, restricting their accuracy during extreme weather transitions.

To validate the relative superiority among the algorithms, a pairwise statistical comparison was conducted using probability estimates of Mean Squared Error (MSE) differentials via Orange's Test & Score module:

- KNN vs. SVM: The probability that KNN yields a higher MSE than SVM is 0.000, while the probability that SVM yields a higher MSE than KNN is 1.000. This definitively confirms that KNN statistically outperforms SVM with near-absolute certainty.
- KNN vs. Random Forest: The probability that KNN yields a higher MSE than Random Forest is 0.804. Although the absolute difference in MSE between KNN (0.467) and Random Forest (0.450) is relatively narrow, this probability indicates that Random Forest maintains a consistent, measurable advantage over KNN across cross-validation folds.
- SVM vs. Random Forest: The probability that SVM yields a higher MSE than Random Forest is 1.000. This reinforces that SVM is consistently the weakest model for this high-resolution regression task.

Overall, this comparative analysis robustly establishes Random Forest as the most optimal algorithm among the three, followed closely by KNN, while SVM is definitively disqualified for this application.

The operational behavior of the algorithms was further examined using the Predictions widget, which generates forecast outputs on unseen data splits, as illustrated in Figure 2.

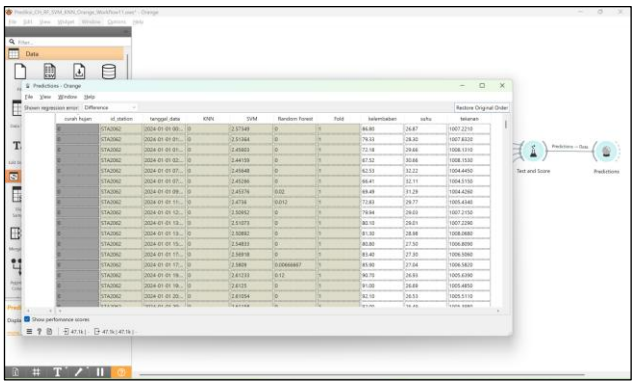

Figure 2. Display of Prediction Results of the Three Models

The visual comparison between predicted and actual rainfall values demonstrates that Random Forest adapts far better to actual data trends, producing very small error residuals that approach zero during non-rain and light-rain intervals. This proves that Random Forest's ensemble bagging mechanism successfully captures local non-linear interactions among humidity, temperature, and pressure. In contrast, KNN and SVM generated higher and rigid error residuals ranging between 2.4 mm and 2.6 mm across multiple observation instances. This indicates that KNN's reliance on localized feature-space Euclidean distance and SVM's kernel constraints lack the structural flexibility required to adapt to sudden micro-scale weather variability.

Despite the mathematical limitations in explaining target variance (as indicated by R^2), the findings provide vital operational insights for institutional stakeholders. For the Agency for Meteorology, Climatology, and Geophysics (BMKG), this study demonstrates that surface-level AWS parameters alone are insufficient for standalone 10-minute precipitation nowcasting; operational deployment must assimilate terrestrial AWS data with weather radar reflectivity or satellite imagery to overcome zero-inflated biases [26]. For the South Tangerang City Government, the Random Forest model—having demonstrated the most stable error rates—can be integrated as an initial data-driven filtering layer within an IoT-based urban flood early warning architecture [26]. By detecting rapid short-term precipitation anomalies, municipal authorities can proactively activate stationary and mobile drainage pumps, adjust retention basin water gates, and deploy emergency mitigation teams to critical flood-prone intersections (such as Ciputat and Pamulang) minutes before surface runoff accumulates to hazardous levels.

VI. CONCLUSION

From the evaluation of the three regression algorithms—Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), Random Forest emerged as the best performer, consistently achieving the lowest RMSE and MAE values among the tested models (0.671 and 0.090, respectively), indicating the smallest absolute prediction error. KNN followed with very competitive performance (RMSE: 0.683; MAE: 0.083), demonstrating comparable predictive ability to Random Forest. In contrast, SVM demonstrated clearly inferior performance, with significantly higher RMSE and MAE values (2.553 and 2.511, respectively), confirming its unsuitability for the characteristics of this dataset. Statistical comparisons between the models also definitively confirmed the superiority of Random Forest and KNN over SVM.

Although Random Forest and KNN effectively minimized the absolute error, a key finding lies in the negative Coefficient of Determination (R^2) values observed across all models (ranging from -0.062 for Random Forest to -14.378 for SVM). These negative R^2 values serve as a strong indicator that the implemented models cannot effectively

explain the variance in the target variable (rainfall), even performing worse than the baseline model that simply predicts the mean of observations. This condition is further exacerbated by the nature of the rainfall data, which often contains zero or very small values, posing challenges for the models in distinguishing between no rainfall (zero) and minimal rainfall. Therefore, although Random Forest emerged as the best choice among the three tested models, its ability to predict 10-minute rainfall using the current dataset and configuration still requires further development to achieve optimal accuracy and reliability.

Practically, the results of this study demonstrate that machine learning approaches can support hydrometeorological disaster mitigation policies. For example, the BMKG (Agency for Meteorology, Climatology, and Geophysics Agency) can utilize the Random Forest model as the basis for a high-resolution rainfall forecasting system, while city governments can integrate this forecast information into flood early warning systems to reduce economic losses, minimize social impacts, and increase the resilience of urban infrastructure.

VII. RECOMMENDATIONS

To overcome the structural limitations identified in this study—specifically the zero-inflated target distribution and negative R^2 values—future research should advance along three strategic directions:

1) Adoption of Deep Learning and Hybrid Architectures for Temporal Data: Standard regression models treat each 10-minute observation independently, disregarding temporal continuity. Future studies should implement sequential Deep Learning models such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU). LSTM networks are explicitly designed to capture long- and short-term temporal dependencies in time-series data, making them significantly superior in modeling the progressive atmospheric buildup leading to a convective rainfall event. Furthermore, hybrid architectures combining RF or XGBoost for feature selection with LSTM for sequential prediction should be explored to maximize predictive stability.
2) Two-Stage (Hurdle) Modeling Approach: To resolve the zero-inflated dataset challenge, a two-stage classification-regression architecture is strongly recommended. The first stage should utilize a classification algorithm (e.g., Logistic Regression or Gradient Boosting) solely to predict the *probability of rainfall occurrence* (rain vs. no-rain). If rain is predicted, the second stage activates a dedicated regression model trained exclusively on non-zero data to estimate exact *rainfall intensity (mm)*.
3) Enrichment of Meteorological Predictors: Relying exclusively on surface temperature, humidity, and air pressure restricts model performance. Future AWS data collection must incorporate additional atmospheric drivers, specifically wind speed and wind direction (to track convective storm cloud advection), solar radiation/sunshine duration, and cloud cover/cloud top temperature from satellite or radar imagery. Incorporating these variables will provide ML algorithms with the thermodynamic and spatial context required to achieve high reliability in operational urban flood forecasting across Indonesian cities.

VIII. THANK YOU

The authors would like to express their deepest gratitude to The Agency for Meteorology, Climatology, and Geophysics (BMKG) for providing the weather data used in this research. This research was supported by Pamulang University, who supported this research. The author also appreciates the valuable input and suggestions from the reviewers that have improved the quality of this paper.

REFERENCES

[1] C. Octavia and K. Marko, "Area vulnerability to flooding in Rangkasbitung District and its surroundings, Lebak Regency, Province of Banten," *E3S Web of Conferences*, vol. 211, p. 02002, Nov. 2020, doi: 10.1051/e3sconf/202021102002.

[2] M. Wang *et al*., "Urban Flooding Risk Assessment in the Rural-Urban Fringe Based on a Bayesian Classifier," *Sustainability*, vol. 15, no. 7, p. 5740, Mar. 2023, doi: 10.3390/su15075740.

[3] S. M. Toufique, S. U. Bhuiyan, A. Lateef, A. Zaman, J. Bin Islam, and D. Z. Karim, "Implementing Machine Learning Techniques to Forecast Floods in Bangladesh," in *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET*, IEEE, Jul. 2024, pp. 1–6. doi: 10.1109/ICECET61485.2024.10698703.

[4] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, Fifth. New Jersey: John Wiley & Sons, 2016.

[5] O. V. Petrochenko, "The problem of flooding and analysis of the ways of its solution," *Environmental safety and natural resources*, vol. 46, no. 2, pp. 5–22, Jun. 2023, doi: 10.32347/2411-4049.2023.2.5-22.

[6] J. Chen, Y. Li, C. Zhang, Y. Tian, and Z. Guo, "Urban Flooding Prediction Method Based on the Combination of LSTM Neural Network and Numerical Model," *Int. J. Environ. Res. Public Health*, vol. 20, no. 2, p. 1043, Jan. 2023, doi: 10.3390/ijerph20021043.

[7] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, no. 14–15, pp. 2627–2636, Aug. 1998, doi: 10.1016/S1352-2310(97)00447-0.

[8] F. Yuan *et al*., "Predicting Road Flooding Risk with Machine Learning Approaches Using Crowdsourced Reports and Fine-grained Traffic Data," Aug. 2021, [Online]. Available: http://arxiv.org/abs/2108.13265

[9] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[11] X. Liu *et al.*, "Influencing Factors and Risk Assessment of Precipitation-Induced Flooding in Zhengzhou, China, Based on Random Forest and XGBoost Algorithms," *Int. J. Environ. Res. Public Health*, vol. 19, no. 24, p. 16544, Dec. 2022, doi: 10.3390/ijerph192416544.

[12] Hartanto, S. Humaidi, E. Frida, M. Sinambela, and N. Ananda, "Evaluation of Meteorological Radar Precipitation Forecast in Banten," in *2023 International Conference on Information*

Technology and Computing (ICITCOM), IEEE, Dec. 2023, pp. 297–300. doi: 10.1109/ICITCOM60176.2023.10442051.

[13] A. F. Radjab, H. Akib, Rifdan, and Jasruddin, "Partnership in Weather Observation using the Crowdsourcing Method," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 499, no. 1, p. 012019, Jun. 2020, doi: 10.1088/1755-1315/499/1/012019.

[14] W. C. Skamarock *et al.*, "A Description of the Advanced Research WRF Version 3," Jun. 2008.

[15] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," *Nature*, vol. 525, no. 7567, pp. 47–55, Sep. 2015, doi: 10.1038/nature14956.

[16] S. Talbi, L. Guerzouli, and S. Fezzai, "Flood susceptibility zonation map using remote sensing and XGboost, Random Forest, Nearest neighbor models in GIS: a case study Tebessa city, Algeria.," Mar. 31, 2023. doi: 10.21203/rs.3.rs-2710595/v1.

[17] Y. Wu, Z. Zhang, X. Qi, W. Hu, and S. Si, "Prediction of flood sensitivity based on Logistic Regression, eXtreme Gradient Boosting, and Random Forest modeling methods," *Water Science & Technology*, vol. 89, no. 10, pp. 2605–2624, May 2024, doi: 10.2166/wst.2024.146.

[18] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004, doi: 10.1023/B:STCO.0000035301.49549.88.

[19] D. Valkenborg, A.-J. Rousseau, M. Geubbelmans, and T. Burzykowski, "Support vector machines," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 164, no. 5, pp. 754–757, Nov. 2023, doi: 10.1016/j.ajodo.2023.08.003.

[20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.

[21] J. Demšar *et al.*, "Orange: Data Mining Toolbox in Python," *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013.

[22] J. Liu *et al.*, "Assessment of Flood Susceptibility Using Support Vector Machine in the Belt and Road Region," May 25, 2021. doi: 10.5194/nhess-2021-80.

[23] I. E. Mfon, M. C. Oguike, S. U. Eteng, and N. M. Etim, "Causes and Effects of Flooding in Nigeria: A Review," *International Journal of Social Science And Human Research*, vol. 05, no. 10, pp. 4526–4533, Oct. 2022, doi: 10.47191/ijsshr/v5-i10-16.

[24] N. Gauhar, S. Das, and K. S. Moury, "Prediction of Flood in Bangladesh using k-Nearest Neighbors Algorithm," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, IEEE, Jan. 2021, pp. 357–361. doi: 10.1109/ICREST51555.2021.9331199.

[25] K. Rhamadania, M. Makhsun, C. Basir, and N. Ananda, "Imputation of Missing Weather Data in Automatic Weather Station Using the GRU Algorithm," *Journal of Applied Informatics and Computing*, vol. 10, no. 2, pp. 1165–1171, Apr. 2026, doi: 10.30871/jaic.v10i2.12301.

[26] L. See, "A review of citizen science and crowdsourcing in applications of pluvial flooding," Feb. 26, 2019, *Frontiers Media S.A.* doi: 10.3389/feart.2019.00044.