khazanah informatika
Jurnal Ilmu Komputer dan Informatika

Flood Prediction Using Machine Learning Model Integrated with Geographical Information System

Muhammad Ricky Perdana Putra [1], Rama Ashari [2], Muhirin [3], Azib Widad Zuhaily Imam [4], Kusrini [5*]
*correspondence: kusrini@amikom.ac.id
[1, 2, 3, 4, 5] Informatics Engineering Department
Universitas Amikom Yogyakarta
Yogyakarta, Indonesia

Abstract- Flooding in Indonesia is still a frequent natural disaster compared to other types of disasters. In addition, the number of flood events also shows an increase every year. This research aims to develop a flood prediction model as a preventive measure as an early warning system and flood risk mitigation management that may occur based on Geographical Information System (GIS). It is expected that areas that have the potential to experience flooding can be more proactive in making preparations before flooding. This prediction model uses a classification type machine learning (ML) algorithm with training data involving rainfall within 12 months. The model evaluation results use two techniques: confusion matrix and K-Fold cross validation and each fold is calculated for accuracy. The K-Nearest Neighbors (KNN) model with a value of K = 31 gets the highest accuracy value of 88.89%, Decision Tree (DT) of 72.22%, and Naive Bayes of 78%. The average accuracy using K-Fold resulted in 89.09% for KNN, 77.12% for DT, and 86.59% for Naive Bayes. By considering these results, this research chose the KNN method to be applied in the prediction model. The code was rewritten in the Flask framework to be used as an API and integrated with Laravel as a Backend platform and Frontend using Bootstrap, JQuery, Axios, and LeafletJS as map visualisation. With this research, it is hoped that it can be one of the solutions in predicting as well as early warning of floods so that it can provide sufficient time for affected residents to make preparations for flooding.

Keywords: flood prediction, machine learning, geographical information system

Article info: *submitted December 28, 2023, revised March 19, 2024, accepted January 8, 2025*

1. Introduction

Floods are the most frequent disaster in Indonesia. Heavy rainfall causes rivers to overflow due to the large amount of water that must be accommodated [1]. Of the total disasters that occurred in Indonesia, around 37% of them were floods [2]. Rain is an unavoidable condition, especially in the rainy season, because high rainfall intensity can cause flooding, so it needs to be watched out for. Overflowing river water that causes puddles in the lowlands from the riverbank is the main cause of flooding.

According to research from the World Risk Report (WRR), Indonesia is ranked second as the country with the second highest risk of natural disasters in the world. Specifically for flood disasters, according to data from the National Disaster Management Agency, the percentage of floods reached 36.50% in 2020. One of the causes of flooding is high rainfall in an area [3]. In addition, environmental damage due to deforestation and reduced water catchment areas also worsen flood conditions [4].

Measuring water surface elevation is very important to detect floods, droughts and normal conditions. With fluctuations in water levels such as lakes or rivers can be a reference for water conditions can be known and anticipate the possibility of floods, droughts or normal conditions [5]. The water level monitoring process is carried out manually, where technicians must walk to the water measurement location, return to the observation office, then record and report the data to the head office and the water gate observation post using HT (Handy Talkie) radio and telephone [6].

In addition, monitoring can be carried out with the Automatic Water Level Recorder (AWLR) using a sensor integrated with the web and there is a visual graph related to the rise and fall of river water elevation [7] and can be integrated with artificial intelligence [8]. Floods are detected at flood detection stations through devices installed in flooded areas. However, it is difficult to detect floods in large areas. With hydrological models, analytical datasets and remote sensing can be used to monitor flood conditions on a large scale, then alternatives can be used with flood maps.

One of the natural factors causing flooding is rainfall intensity, which can be used as a basis for predicting the occurrence of flooding in a particular area [9]. Areas with high rainfall are more prone to flooding. Therefore, a system is needed to predict flooding so that information or early warnings of flooding can be conveyed to local residents automatically and in real time so that they can be aware of possible flooding disasters. One way to provide information to residents of an area, especially those bordering rivers, is through the internet.

With the increasingly massive development of artificial intelligence (AI) technology, it has influenced the development of map-based flood detection tools known as GeoAI [10]. An example of a type of AI for flood prediction is Machine Learning

(ML), which can learn data patterns and produce predictions based on the latest data [11]. This provides benefits to people living in flood-prone areas because AI can provide predictive information and people can prepare themselves for mitigation plans and perhaps evacuate to safety.

Previous studies were conducted with various datasets and methods, and each has its own advantages and disadvantages. Hulwana et al. used the 2019-2021 West Kalimantan rainfall dataset for training and 2022 data for testing [12]. The attributes used are temperature, humidity, rainfall, wind direction, and duration of sunlight. The model with the Deep Learning H2O algorithm on hidden layers 100, 50, 100 which produces an accuracy value of 90.54% and Root Mean Squared Error (RMSE) 0.087. The shortcomings of this study are that the data pre-processing stage only uses one technique, namely deleting rows that have null data, there is no confusion matrix image, and the evaluation of the algorithm model has not used the K-Fold cross validation technique.

Frenica et al. conducted research because the South Sumatra region was hit by repeated floods in the period 2022 to 2023, so as a preventive measure, a flood prediction model was built using the Support Vector Machine (SVM) algorithm with a dataset from Kaggle.com which groups floods into three statuses, namely safe, alert, and danger with a total of 16,272 rows [13]. The attributes used are water elevation, water discharge, rainfall, temperature, humidity. Based on the evaluation of the model, it produces a value of 1.0 for accuracy, precision, recall, and F1-Score. However, the drawback of this study is that it cannot predict floods for the future because it does not have a time attribute.

Fitrianah et al conducted a comparison of algorithms used for flood prediction, namely SVM, Decision Tree C5.0, and Naïve Bayes [10]. The dataset used came from the website of the Meteorology, Climatology, and Geophysics Agency (BMKG) and the Central Statistics Agency (BPS). The variables or features used include rainfall, sub-district, water discharge, area, duration of rain, and population density. The results of this study are that the SVM and C5.0 algorithms obtained an accuracy value of 93.75% and Naïve Bayes of 81.25. The disadvantage of this study is that it does not explain the pre-processing stage carried out because this stage also has an urgency that can affect the accuracy value of the model built [11]. In addition, the visual of the confusion matrix is not displayed.

Altunkaynak et al. developed a hybrid fuzzy based model and implemented it in a geographic information system (GIS) [14]. Based on observations of the Ayamama River basin in Istanbul, Turkey. GIS is used as an early warning system and flood hazard management that usually uses a conventional hydrological approach to artificial intelligence-based. The dataset used is the Istanbul Metropolitan Municipality Water Works and Canalization Administration (ISKI) from 1987 to 2015. However, this study does not use machine learning (ML) algorithms for flood prediction.

As a more effective and efficient preventive measure and as an early warning, management, and flood mitigation plan, it can be done with flood prediction with time series-based ML technology. As a form of contribution, the research conducted will focus on three things, namely pre-processing, model development with KNN, and model implementation to the web. KNN was chosen because it is simple, fast execution time, robustness to various types of data, easy to integrate, and adaptive to various scenarios [15] and has the potential to improve its accuracy [16].

Another contribution is that this study combines the results of the ML model that has been built to be integrated into the GIS web. To develop it, several tools are needed such as the Python Flask framework, PHP Laravel, JQuery, CSS Bootstrap, and LeafletJS. A more detailed explanation will be presented in the methodology section. This aims to increase the user experience in accessing prediction applications compared to just entering data and outputting prediction results alone. So it is better to have an attractive and interactive map visual.

Based on the four research studies that have been explained above, this study has two main objectives, namely experimenting with building the best performing ML model and integrating it into the GIS web. The web was chosen because of its flexibility and can be accessed on all platforms such as Android, Windows, Mac, Linux, and so on. The web application will be disseminated to the public so that it can be accessed anytime and anywhere via the internet. This aims to ensure that residents can be prepared and alert to possible flood disasters.

2. Methods

The technology used to develop this web application is from the frontend side there is a JQuery library to simplify the Vanilla Javascript syntax. Axios library for request management to the backend to make it simpler. Third-party Leaflet.JS as a dynamic and interactive map visualization whose data comes from model prediction results. The latest CSS Bootstrap framework version 5.3 equipped with Popper.JS for display interactivity and as styling. The backend side uses the PHP Laravel framework version 10. The model is placed on the Flask Python framework and the interaction between Laravel and Flask uses the API.

First, the research was conducted by building a model experimentally or trial and error to find the most effective algorithm or combination based on previous research. With the help of Google Colab using the KNN, Decision Tree, and Naïve Bayes algorithms, then choosing the model with the highest accuracy value among the three. After that, the code from Google Colab will be rewritten in Flask to be used as a prediction Application Programming Interface (API) that will be used through the Laravel framework. The research flow can be seen in Figure 1.

The dataset used has been made publicly available through the page https://www.kaggle.com/code/mukulthakur177/flood-prediction-model/input. The file on the page can be downloaded in CSV format. The contents of the dataset consist of monthly rainfall in millimeters and flood conditions or not in the Kerala region, India from 1900 to 2018. The number of rows in the dataset is 118 with features such as Subdivision, Year, Jan, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, Annual Rainfall, and Flood which can be seen in the table below Figure 2.

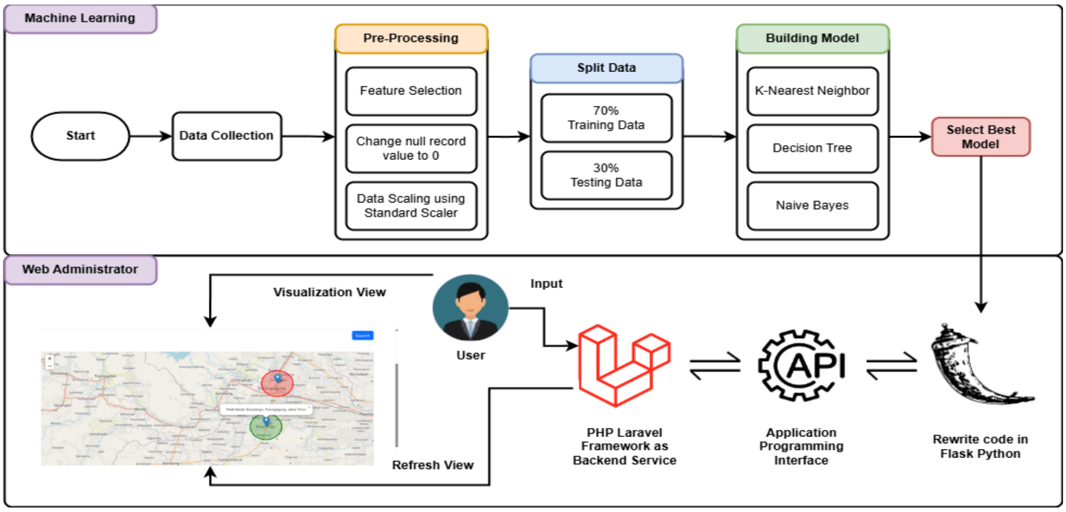

Figure 1. Research Flow

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL RAINFALL	FLOODS
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6	YES
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6	YES
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2	YES
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	YES
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	NO

Figure 2. Raw Dataset

Next, data pre-processing is carried out in three stages, namely (1) checking whether there are empty or unfilled columns, if there are empty columns then they must be filled with the number 0, (2) selecting the features used in Figure 3, (3) scaling the data with Standard Scaler. The reason is that it can eliminate averages, adjust data variations, is suitable for data-sensitive algorithms such as KNN and reduces the influence of outliers [17]. The results of the dataset after scaling are shown in Figure 3 below.

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	FLOODS
0	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	YES
1	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	YES
2	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	YES
3	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	YES
4	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	NO

Figure 3. Dataset After Scaling with Standard Scaler

Next, determine the independent variables which include 13 features, namely Year, Jan, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, and one feature for the dependent variable, namely Flood, which is then called the variable x and y. From the data that has been determined, a separation is carried out between training data and test data with a percentage of 70% and 30%, resulting in new variables, namely x_train, x_test, y_train and y_test. This separation process aims to ensure that the model can be tested with data that has never been seen before so that model evaluation becomes more objective.

The model training stage utilizes the ML Python library, namely Scikit-Learn. This aims to speed up the process because it contains functions that can be used directly. After the training process is complete and produces three models to be compared between KNN, DT, and NB, the next step is testing using the confusion matrix and K-Fold cross validation. The focus of testing is the accuracy value which is a metric to measure how well the predictive model correctly classifies data.

The performance value is obtained from the iteration of K-Fold cross validation with K = 10, which means that the training data is divided into ten subsets with nine subsets as training data and one subset as test data. K-Fold is performed ten times with different test subsets. The accuracy of each iteration will be calculated. From the average value, one best model is selected to be implemented into a web application to predict flood events based on monthly rainfall.

Implementation to the web requires a framework from Python which is known for being simple and fast, namely Flask. Flask is used as a place to put the best previous ML models. So, Laravel will send input data from the user to Flask via AP to be predicted by the ML model. The prediction results will be sent back to Laravel and then forwarded to the Frontend view and visualized using LeafletJS for flood area mapping. This approach allows integration between Flask as a predictive Backend and Laravel as the main platform while Leaflet provides an interactive and informative display for users.

3. Result

The cleaned dataset through the pre-processing stage includes (1) replacing missing data with zero, this is done to complete and maintain data consistency so as to avoid crashes. (2) selecting the features used including year, rainfall from 1st to 12th month, annual rainfall, and prediction label, (3) performing Standard Scaler. The three pre-processing stages are expected to ensure the quality of the data used to build the prediction model, ensure relevance, prevent bias, and more accurate data patterns.

Next is building a model using the KNN algorithm. To optimize the model that is built, what must be done is to find the most optimal k value. The k value is the distance of the nearest neighboring circle. The greater the k value, the greater the neighboring circle and the greater the possibility of prediction. So it is better to use an odd k value to avoid deadlock [18]. The results of data training to determine the k value are presented in Figure 4 below.

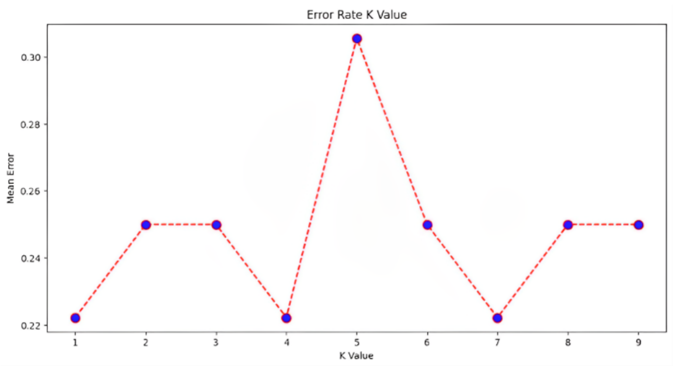

Figure 4. Error *k* Rate Value

It doesn't stop there, to find the most optimal k value, GridSearchCV is used. This technique is used to find the best combination of hyperparameters for a particular model by utilizing grid search and cross validation [19]. KNN is suitable for hyperparameter tuning with GridSearchCV due to the sensitivity of the k value, model simplicity, optimal performance evaluation, and easy implementation [20]. The results of GridSearchCV show that the most optimal k value is 31 and this value is used.

The results of training and testing the three models, namely (1) KNN with a value of k = 31 obtained an accuracy value of 88.89%, (2) Decision Tree obtained a value of 72.22%, (3) Naïve Bayes obtained a value of 78%. Of the three algorithms, the one that obtained the highest accuracy value was KNN. Furthermore, an evaluation was carried out using the K-Fold cross validation technique. After all folds have been trained and tested, the accuracy value will be averaged to see the stability and resilience of the model in dealing with various types of data. The results of the confusion matrix visualization of the KNN model are presented in Figure 5 and the results of the K-Fold evaluation are presented in Table 1 below.

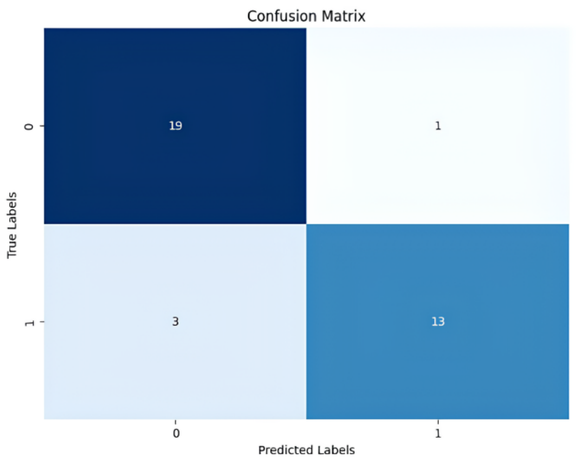

Figure 5. Confusion Matrix KNN

Tabel 1 K-Fold Cross Validation

Model/Fold	KNN	Decision Tree	Naïve Bayes
k=1	100	75.00	100
k=2	83.33	75.00	75.00
k=3	100	83.33	83.33
k=4	75.00	50.00	83.33
k=5	83.33	91.67	91.67
k=6	91.67	91.67	83.33
k=7	75.00	75.00	75.00
k=8	91.67	75.00	83.33
k=9	100	90.91	100
k=10	90.91	63.64	90.91
Average	89.09	77.12	86.59

In accordance with the research flow, the Google Colab code for building the KNN model was rewritten in the Flask Python framework to be used as an API to be used in Laravel and the prediction results were stored in a MySQL database. In the web application, there are two user roles, namely visitors and administrators. There are three features for visitors, namely location data search (district, city/regency, and province), search by year (prediction), and visualization in the form of a map. When the user opens the web address https://flood-prediction.larachain.my.id, the following page will appear for the roles of visitors and administrators.

The main page is shown in Figure 6 below. It has several things, namely a button to login, a form for searching by sub-district, city/district, and province and year, then there is a search button based on user input. At the bottom there is a map that will adjust to the results of user input. Data processing on the frontend side uses Axios so that it does not require page reloading for data rendering because data exchange is done in the background and displayed directly through document object model (DOM) modification.

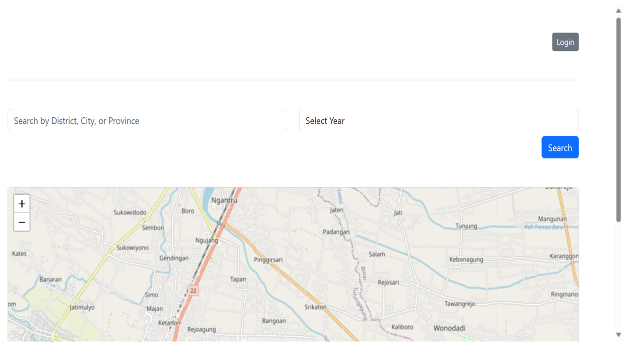

Figure 6. Visitor View

Next, for the login page as in Figure 7 below, only administrators can access it. There are two inputs, namely postal mail and password. If authentication is successful, the admin will be directed to the homepage, if it fails, it will be directed back to the login page and given information about the cause of the failed login, such as postal mail not registered or wrong password. For further research, it can provide a security mechanism such as recording the user's IP address when there is a login attempt and after several failed attempts, the account can be temporarily locked.

After successfully logging in, the administrator can enter rainfall data and the ML model will group it into binary, namely flood and not flood. In this display there is a third-party library integrated with Laravel, namely Yajra Datatable to process and display data on the server side in order to save resources. The columns displayed are numbers (index, sorted from number 1), district/city, year, and 12 months, as well as the situation based on the results of ML predictions and actions to edit and delete data.

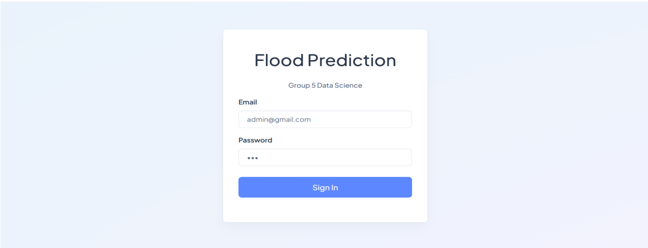

Figure 7. Administrator Login View

The default function of Yajra Datatable is pagination, the number of data displayed with a default of 10 data and can be changed to 25, 50, and 100 data. Then there is a column for searching in all columns and rows of data that may match, this aims to facilitate data checking and management. On the upper right side there is a default admin profile photo and when clicked a button to logout will appear. The add data button under the profile photo when clicked will direct to the data input page. The appearance of this feature is presented in Figure 8 below.

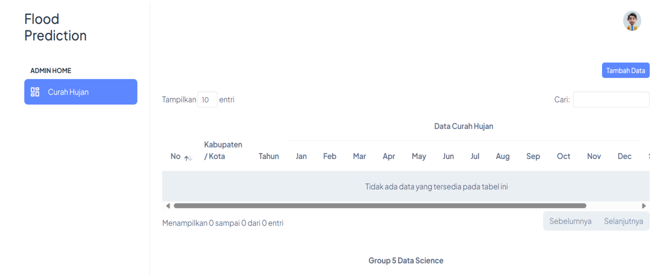

Figure 8. After Login Display

The formular display for data input is adjusted based on the dataset column and ease of user experience (UX). After everything is filled in, the rainfall data will be sent to Flask to predict whether or not there will be a flood. The results are sent back to Laravel to be stored in the database. In addition, there are several fields added to support visualization in LeafletJS, namely Longitude, Latitude, and location name. The input form is depicted in Figure 9 below.

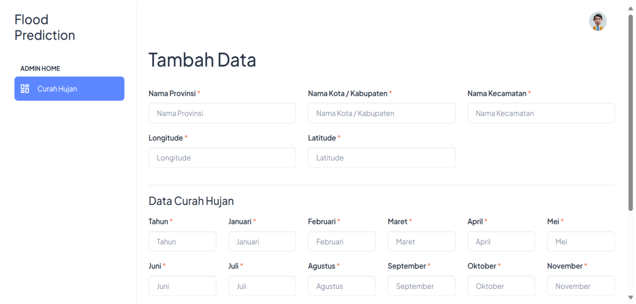

Figure 9. Input Form

Longitude and latitude are used to obtain the coordinates of flood points and visualized within a certain radius. There are two colors in the radius, namely red means the area is potentially flooded and green means the area is not potentially flooded. To add to the experience, when a point or radius is clicked, it will display the location label so that it is clearer. An example of the visualization results is presented in Figure 10 below.

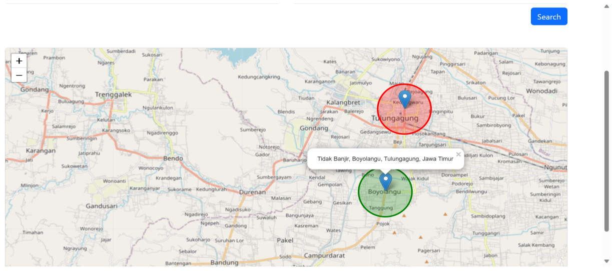

Figure 10. Main View After Dot

4. Discussion

It has been mentioned in the previous sub-chapter that this web application uses a Python-based ML model to predict whether a location point is submerged or not. Of the three algorithms used, namely KNN, Naive Bayes, and Decision Tree, the algorithm chosen and used in this study is KNN with K = 31 which is the model with the highest accuracy of 88.89%. Compared to the other 2 models, namely DT with an accuracy of 72.22% and Naive Bayes with an accuracy of 78%.

Comparison of previous research conducted by Hulwana et al. referring to the 2019-2022 West Kalimantan rainfall dataset has the weakness of only carrying out one pre-processing stage, namely deleting missing data rows. While this research carries out three sub-stages of pre-processing, namely deleting missing data, selecting common features with the hope that the system built can be used widely, and scaling data with Standard Scaler.

Then, another deficiency that is trying to be covered in this study is the visualization of the evaluation results based on confusion matrix and the results of the K-Fold cross validation table, and the failure to implement the prediction results in the map. Even so, the research that was conducted still has shortcomings in the accuracy of the model which is still lacking by 1.65%. This can be a further suggestion to add optimization techniques such as Particle Swarm Optimization and ensemble learning techniques by combining two or more algorithms.

Frenerica et al.'s research in the South Sumatra region in the period 2022 to 2023 as a warning to the public with predictions using SVM. The drawback is the absence of time series data so that predictions cannot be made for a certain period of time. Meanwhile, the research conducted uses a time series of monthly rainfall over a full year for predictions. This is an advantage in analyzing rainfall patterns over time so that more accurate and relevant predictions can be made based on historical trends.

Fitrianah et al. compared three ML algorithms for flood prediction including SVM, Decision Tree, and Naïve Bayes. There is one unique variable compared to previous research and the research conducted, namely population density. However, this study still does not explain the pre-processing stage and the absence of visualization of the confusion matrix results. The research conducted attempts to cover these shortcomings by conducting three sub-pre-processing stages and displaying the visualization of the KNN confusion matrix which is the model with the highest performance in this research.

Finally, research by Altunkaynak et al, developed a GIS based on a hybrid fuzzy-based model. Conceptually, this research is almost the same as the research conducted because it conducted training on data that had been collected before being visualized in GIS. However, this research uses hybrid fuzzy with a focus on decision making based on linguistic rules, different from the research conducted, which is to produce an ML model to predict new data. Both have the same integration into GIS for more interesting and interactive information.

5. Conclusion

The test results of the three algorithms KNN, Decision Tree, and Naïve Bayes with confusion matrix obtained a value of 88.89% for KNN, a value of 72.22% for DT, and a value of 78% for Naïve Bayes. Meanwhile, the evaluation using K-Fold cross validation produced an average accuracy value of 89.09% for KNN, 77.12 for DT, and 86.59 for Naïve Bayes. The model with the best performance, namely KNN, was rewritten on the Flask Python framework as an API. Then, a geographic information system (GIS) was developed with Laravel as a backend to consume the API from Flask. Bootstrap, JQuery, and Axios as frontends plus LeafletJS as flood map visualization.

This research also contributes to expanding the scope of the use of AI-based technology for early warning and disaster mitigation management, thus providing a positive contribution to the development of ML models integrated with GIS development to predict flood disasters. Because most of the research related to this topic has a specific purpose between only building ML models or developing GIS only. While this research seeks to collaborate the two with the hope that it can be used directly by the community.

This research still has shortcomings, one of which is in the performance of the model which in further research can be analyzed more deeply, can implement optimization techniques such as PSO, Komodor Mlipir Algorithm (KMA), and Ant Colony. In addition, it can experiment using the concept of ensemble learning, namely combining more than one algorithm with the aim of increasing model accuracy. On the visual side of the frontend, it can add user experience with a UI/UX design that is pleasing to the eye and real-time notifications.

Reference

[1] M. Seprianto, M. Anggo, L. Harudu, and S. Aldiansyah, "Pemetaan Daerah Potensi Rawan Banjir Menggunakan Metode Overlay," vol. 9, no. 4, 2024.

[2] B. Banjir, P. dan Pengendalian Pemanfaatan Ruang, P. U. dan Pengendalian Pemanfaatan Ruang Berdasarkan Penataan Ruang dan RUU Cipta Kerja, S. Nurhayati Qodriyatun, and P. R. Penelitian Badan Keahlian DPR Jl Gatot Subroto, "Sri Nurhayati Qodriyatun", doi: 10.22212/aspirasi.v11i1.1590.

[3] H. Rakuasa, D. A. Sihasale, M. C. Mehdila, and A. P. Wlary, "Analisis Spasial Tingkat Kerawanan Banjir di Kecamatan Teluk Ambon Baguala, Kota Ambon," *Jurnal Geosains dan Remote Sensing*, vol. 3, no. 2, pp. 60–69, Nov. 2022, doi: 10.23960/jgrs.2022.v3i2.80.

[4] H. Wahyuni and S. Suranto, "Dampak Deforestasi Hutan Skala Besar terhadap Pemanasan Global di Indonesia," *JIIP: Jurnal Ilmiah Ilmu Pemerintahan*, vol. 6, no. 1, pp. 148–162, Mar. 2021, doi: 10.14710/jiip.v6i1.10083.

[5] G. Ifeoma and G. Onyekachi, "Development of a low-cost automatic water level monitoring system," 2019. [Online]. Available: http://www.cigrjournal.org

[6] A. Rinny Asnaning and S. Dwi Putra, *Flood Early Warning System Using Cognitive Artificial Intelligence: The Design of AWLR Sensor.*

[7] W. A. Permadi, H. P. K. Habibi, and M. H. H. Putri, "Pengembangan Automatic Water Level Recorder (AWLR) Berbasis IoT Sebagai Alat Mitigasi Resiko Potensi Bencana Banjir di Kota Bontang," *PoliGrid*, vol. 2, no. 1, p. 30, Jun. 2021, doi: 10.46964/poligrid.v2i1.725.

[8] T. A. Pianto *et al.*, "Design and implementation of an IoT-based Automatic Water Level Recorder (AWLR) as a tool to mitigate the risk of tidal flood disasters in the coastal area of Jakarta," in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, 2023. doi: 10.1088/1755-1315/1266/1/012061.

[9] S. Puttinaovarat and P. Horkaew, "Flood Forecasting System Based on Integrated Big and Crowdsource Data by Using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 5885–5905, 2020, doi: 10.1109/ACCESS.2019.2963819.

[10] A. Muin, H. Rakuasa, and K. Kunci, "Pemanfaat Geographic Artificial Intelligence (Geo-AI) Untuk Identifikasi Daerah Rawan Banjir Di Kota Ambon," *Gudang Jurnal Multidisiplin Ilmu*, pp. 58–63, 2023, doi: 10.59435/gjmi.v1i2.24.

[11] D. Salsabiil Susanto and N. Deny Hendrawan, "Perancangan Kecerdasan Buatan untuk Analisis dan Segmentasi Pemetaan Area Terdampak Banjir Berbasis Computer Vision," *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, vol. 7, no. 3, 2024.

[12] H. Sharfina, P. Y. Utami, and I. Fakhruzi, "Prediksi Bencana Banjir Menggunakan Algoritma Deep Learning H2O Berdasarkan Data Curah Hujan," 2023. [Online]. Available: http://jurnal.mdp.ac.id

[13] A. Frenica and S. Soim, "Implementasi Algoritma Support Vector Machine (SVM) untuk Deteksi Banjir," vol. 8, no. 2, p. 2023.

[14] A. Altunkaynak and H. Bizimana, "Prediction of flood hazard map based on hybrid fuzzy geographic information system and its application for Ayamama watershed," 2020.

[15] S. Liu, R. Liu, and N. Tan, "A spatial improved-knn-based flood inundation risk framework for urban tourism under two rainfall scenarios," *Sustainability (Switzerland)*, vol. 13, no. 5, pp. 1–19, Mar. 2021, doi: 10.3390/su13052859.

[16] S. V. Razavi-Termeh, A. Sadeghi-Niaraki, S. Razavi, and S. M. Choi, "Enhancing flood-prone area mapping: fine-tuning the K-nearest neighbors (KNN) algorithm for spatial modelling," *Int J Digit Earth*, vol. 17, no. 1, 2024, doi: 10.1080/17538947.2024.2311325.

[17] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies (Basel)*, vol. 9, no. 3, Sep. 2021, doi: 10.3390/technologies9030052.

[18] A. Sanjaya and T. Wahyana, "PENERAPAN METODE K-NEAREST NEIGHBOUR UNTUK SISTEM PREDIKSI KELULUSAN SISWA MTS NURUL MUSLIMIN BERBASIS WEBSITE," *Journal Transformation of Mandalika*, vol. 3, no. 1, 2022, [Online]. Available: http://ojs.cahayamandalika.com/index.php/jtm/issue/archive

[19] W. Nugraha and A. Sasongko, "Hyperparameter Tuning on Classification Algorithm with Grid Search," *SISTEMASI*, vol. 11, no. 2, p. 391, May 2022, doi: 10.32520/stmsi.v11i2.1750.

[20] E. E. Fitriani and W. Yustanti, "Perbandingan Kinerja Metode Problem Transformation-KNN dan Algorithm Adaptation-KNN pada Klasifikasi Multi-Label Pertanyaan Kotakode," 2022.