FLOOD PREDICTION USING MACHINE LEARNING

Abdalla Alamen and Wyatt Clausen

College of Science, Engineering, and Technology, Minnesota State University, Mankato, USA

ABSTRACT

Floods are destructive and frequent natural disasters. Because of this, machine learning models have been developed in an attempt to predict flooding. Furthermore, this project aims to review a variety of methods such as Long Short-Term Memory (LSTM), LightGBM, Multilayer Perceptron, Support Vector Machine, and Random Forests in their ability to predict floods using a multivariate dataset of historical flood data from Bangladesh (1949-2014) and a time-series dataset for the Minnesota River (2019-2025). The performance metrics of interest for this project were accuracy, precision, recall, F1-Score, Mean Square Error (MSE) and its root (RMSE), Nash-Sutcliffe Efficiency (NSE), and Kling-Gupta Efficiency (KGE). In addition, confusion matrices and ROC curves were developed in order to judge model performance. From this project, the LightGBM model worked best for the Bangladesh data while the LSTM worked best for the time-series data. In addition, the most important features for the LightGBM model were rainfall, recording location, and year.

KEYWORDS

Flood Prediction, LSTM, LightGBM, Machine Learning

1. INTRODUCTION

Among the most destructive and frequent natural disasters are floods. Floods cause a huge and extensive loss of life, infrastructure, and disrupt countries or regions socio-economic systems to be disrupted [2]. According to the United Nations Office for Disaster Risk Reduction (UNDRR), floods have affected billions of people in the past twenty years all over the world. Such natural disasters driven by climate change at its current rate of occurrence and its intensity highlight an urgent need for systems that alert about floods. Previously, flood warning systems used in the past ten years relied on physical apparatuses such as hydrological and hydraulic models. However, such systems can be less accurate due to their constant need for calibration and the input of high-resolution data, which causes issues in regions where data is limited.

On the other hand, such alert systems have seen a significant change due to the advancement of Machine Learning (ML), which has provided a good platform for such systems to function by providing complex, nonlinear relationships between meteorological and hydrological variables. However, such an advancement has its downsides from challenges and limitations in selecting the most suitable approach for the specific hydrological contexts of the country or region, creating gaps in previous research which this research aims to fill.

Regardless of the promising results of machine learning-based flood prediction systems, some challenges remain unaddressed in existing studies. Flood datasets are often imbalanced with non-flood events occurring more frequently than flood events, which can bias the model performance. Additionally, model performance is strongly influenced by the data's structure and regional and hydrological characteristics, which can raise concerns about generalization across different

 DOI:10.5121/ijci.2026.150105

climates and geographical regions. Motivated by these challenges, this study aims to systematically evaluate and compare different machine learning models across two distinct datasets, one multivariate dataset from Bangladesh and one time-series river height dataset from the Minnesota River. This research aims to conduct a comparative assessment of model performance under imbalanced conditions using multiple evaluation metrics, an analysis of model performance based on data characteristics, and a comparison of different datasets for different flood occurrences to adapt the model to the collected data and an evaluation of feature importance to improve the reliability of flood prediction models.

2. RELATED WORK

Previous research has built its findings extensively on physical-based and data-driven approaches, yet as mentioned before, previous models require constant calibration and high-resolution data, as well as computational resources to process the data provided [3][4]. Such approaches have limited their applicability in certain operational contexts, which drove recent research to rely heavily on ML methods to overcome such challenges and create methods for flood prediction.

In some of the literature reviewed, such as Ghorpade et al. [3], several ML models were used as references for flood forecasting by studying the algorithms used, including but not limited to Decision Trees, Linear Regression, Support Vector Machines (SVM), Artificial Neural Networks (ANNs), and ensemble methods. Some findings indicated that Long Short-Term Memory (LSTM) networks were able to identify nonlinear hydrological relationships far better than traditional statistical approaches. While Mosavi et al. [4] identified several other ML models used in flood prediction that used hybrid and ensemble algorithms, highlighting their effectiveness, such findings concluded that merging several algorithms with other physical models further improves the robustness, accuracy, and generalization of the initial findings of the intended model.

In addition, flood forecasting models such as the ones noted above have been reviewed by Akinsoji et al. [7], and the authors note that optimization algorithms such as Grasshopper Optimization and Differential Evolution were used to refine model parameters. Next, Maspo et al. [8] performed another review of flood prediction models and discovered that the most common performance metrics used in studies include Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the correlation coefficient (R^2). Finally, Kruti et al. [9] developed three models for flood prediction, namely Decision Tree, Random Forest, and Gradient Boosting, and identified the Decision Tree as the most accurate.

Nevo et al. [2] experimented with the operational implementation of ML-based systems using Google's end-to-end flood forecasting system in India and Bangladesh as a reference for their research. Google used LSTM and a novel inundation model stage forecasting approach and included two techniques: thresholding and manifold. The combination of both the models and approaches resulted in improved operational accuracy, alongside the manifold approach providing computational support. This combination proved more effective than physics-based hydraulic simulations. In another study by Nevo et al. [5], the authors underlined the deployment of water-level-based hydrological and morphological inundation models that showed high accuracy and low data requirements, demonstrating high scalability for flood forecasting.

Following the previous study where Bangladesh was used as an example, Syeed et al. [4] utilized Bangladesh rainfall data from 34 meteorological stations and compared Binary Logistic Regression, Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), and Decision Tree Classifier (DTC). This resulted in a high accuracy rating of 86.7% achieved by the logistic

regression model by using only simple classifiers in a data-limited region for operational flood forecasting. On the other hand, the ML4FF framework was introduced by Soares et al. [6] for flash flood forecasting in Brazil. This framework further expanded its scope to 34 ML methods across 11 algorithmic classes, including deep learning, applied to the Bangalas River watershed, which showcased the power and efficiency of the LSTM model. This highlighted the effectiveness of automated model selection for flash flood prediction and hyperparameter optimization in complex urban watersheds.

Such research proves that traditional approaches are no longer able to provide the same accuracy, scalability, and operational feasibility when compared to ML methods for improving flood forecasting. Nevertheless, challenges and gaps continue to persist in generalization across diverse hydrological contexts, integration of heterogeneous data sources, and balancing prediction accuracy with computational efficiency. This research further builds on these findings by identifying models that have not been tested yet, while also experimenting with different algorithms that can enhance accuracy and credibility of data.

3. METHODOLOGY

3.1. Datasets

For this project, two datasets were used. The first dataset is relevant to Minnesota in that it contains time-series data from the Minnesota River. The dataset recorded river height in feet, collected by the US Geological Survey at Mankato, Minnesota, from 2019 to 2025 every 15 minutes. A chart containing this time-series data is shown in Figure 1 below. In addition to that dataset, the second dataset comes from Bangladesh and contains more data. To be specific, the Bangladesh dataset contains 20,544 observations and 18 features, which are shown in Table 1 below. In addition, the dataset was recorded from 1949 to 2014. The features were mostly numeric, and there were no missing data values. Finally, the dataset contained 16,412 non-flood observations and 4,132 flood observations, implying a moderate imbalance in the dataset, which may make the models biased toward predicting non-floods, particularly for Random Forests and SVMs. Because of this, performance metrics beyond accuracy were used to measure model performance.

Table 1 . Bangladesh Data features

Name	Type	Unit
Station Name	Categorical	
Year	Numerical	
Month	Categorical	
Max Temp	Numerical	Celsius
Min Temp	Numerical	Celsius
Rainfall	Numerical	cm
Relative Humidity	Numerical	Percentage
Wind Speed	Numerical	m/s
Cloud Coverage	Numerical	Okta
Bright Sunshine	Numerical	Hours/Day
Station Number	Categorical	
X Coordinate	Numerical	
Y Coordinate	Numerical	
Latitude	Numerical	
Longitude	Numerical	
Altitude	Numerical	m
Period	Date	Year.Month
Flood?	Binary	1 = Yes, Blank = No

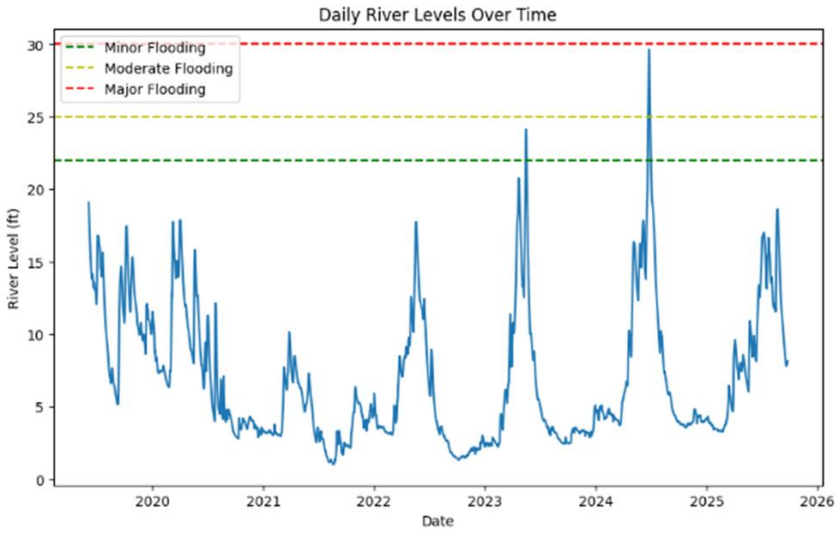

Fig. 1. Minnesota River Height

3.2. Data Processing

After identifying the datasets, the data must be preprocessed for the models. First, for the Minnesota River data, since it was recorded every 15 minutes, the daily average was taken. Next, the blank values for the Flood feature in the Bangladesh data were filled with zeros in order to make it a binary variable. Afterwards, the features: Station Number, Latitude, Longitude, and Period were removed since their roles are satisfied by other variables. Next, the remaining categorical variables were encoded with a Label Encoder to replace the categories with numbers. Next, the numerical features were scaled with a Standard Scaler for the Bangladesh data and Min-Max Normalization for the Minnesota data. Finally, the data was split into a training set with 80% of the data and a test set with the remaining 20%. For the Bangladesh data, this yielded a training dataset containing 16,435 observations and a test dataset containing 4,109 observations. In addition, since the Minnesota dataset is a time series, it was not shuffled to avoid data leakage.

3.3. Models

After preparing the data, several models were used for flood prediction and classification. The first of these models is a time series model such as Long Short-Term Memory (LSTM) that are good for time series forecasting and capturing temporal flood patterns. The next model of interest is gradient boosting with LightGBM which handles tabular data efficiently and captures complex feature interactions. Afterwards, there is a Multilayer Perceptron (MLP) called MLPRegressor which learns nonlinear relationships between factors. Then, there are support vector machines with a variety of kernels that can be used for flood classification. Finally, there are Random Forests which reduce overfitting and work well with mixed datasets.

3.4. Performance Metrics

Finally, after creating the models with training data and making estimations on the test data, the validity of the models was examined with various performance metrics. The first performance metrics of interest used with classifiers are accuracy, precision, recall, and the F-1 score. In addition to these metrics, a confusion matrix can be used with classifiers to give a detailed breakdown of correct versus incorrect predictions while the ROC curve can be used to determine how well the models can distinguish between positive and negative cases. For the time series

models, MSE and its root (RMSE) will be the metrics of interest since they are used with continuous data. The last two performance metrics are more related to hydrology and are the Nash-Sutcliffe Efficiency (NSE) which shows how well predictions match observations and the Kling-Gupta Efficiency (KGE) which assesses correlation, bias, and variability.

4. RESULTS AND DISCUSSION

Using the Minnesota River data, the LSTM model produced a test RMSE of 0.017, a test KGE of 0.993, and a test NSE of 0.993. With a low RMSE and high KGE and NSE, this model was accurate in predicting time-series data. In addition, the LightGBM model generated a test RMSE of 0.037, a test KGE of 0.937, and a test NSE of 0.957. While these performance metrics are slightly greater or lower than those from the LSTM model, the LightGBM model was still accurate with the data. Due to the better performance metrics, it appears that the LSTM model is the best one for the Minnesota River data. Furthermore, other models were not used with this data because they focused on classification rather than regression.

As for the Bangladesh dataset, all of the proposed models were utilized and trained on the data. The following paragraphs show the results when the models were tested with the test data.

Table II . performance comparison across all models.

Model	Accuracy	Precision	Recall	F1-score	MSE	RMSE	NSE	KGE
Random Forest	0.9754	0.9437	0.9334	0.9385	0.0246	0.1568	0.8470	0.9223
LightGBM	0.9786	0.9467	0.9467	0.9467	0.0214	0.1463	0.8667	0.9333
SVM	0.9628	0.9129	0.9007	0.9068	0.0372	0.1930	0.7682	0.8827
MLP	0.9706	0.9201	0.9346	0.9273	0.0294	0.1716	0.8167	0.9074
LSTM	0.9611	0.9142	0.8898	0.9018	0.0389	0.1973	0.7576	0.8744

The model with the highest overall performance was LightGBM with an accuracy score of 97.86%, a precision score of 94.67%, and an F1-score of 94.67%.. This was followed by the next best performing model which was Random Forest with an accuracy of 97.81% and a similar precision and F1-score, which indicates that tree-based models have proven to be the most effective for such flood indication systems.

Furthermore, other models such as MLP demonstrated their ability to capture nonlinear relationships with results that competed with other models with an accuracy of 97.06% and an F1-score of 90.18%. On the other hand, the last two models which are SVM and LSTM that have shown good generalization ability despite their differences, have achieved a lower accuracy of approximately 96.3% than other models.

Lastly, the LightGBM model showed the lowest RMSE (0.1463) in terms of error-based metrics as well as the highest efficiency values (NSE=0.8667, KGE=0.9333). These results demonstrate that the proposed models have shown robust predictive reliability and consistency.

In Figures 2-6, the confusion matrices for all the models are presented and this is where a clearer picture emerges on a comparison basis for classification performance. The figures show the distributions of true positives, true negatives, false positives, and false negatives, providing the ability to visually assess how well each model can identify flood and non-flood events. Agreeing with the performance metrics, LightGBM has achieved the best results of all models, resulting in only 44 false negatives and 44 false positives which is approximately 2% of misclassifications, indicating a balanced ability to detect both classes. Likewise, Random Forest showed similar

performance, with slightly higher false negatives. The MLP model has also performed well, maintaining a low number of misclassified events even though it has 50% higher false-positive values in comparison to LightGBM. Lastly, the SVM and LSTM models showed higher false-positive and false-negative values when compared to the previous models. LSTM, in particular, has missed more actual flood cases due to the non-sequential structure of the Bangladesh dataset. Outside of LightGBM and MLP, the models predicted more false negatives than false positives. This could be due to the imbalanced nature of the dataset that was noted in Section 3.1. Overall, the confusion matrices show that tree-based ensemble models provide the most reliable and stable classification, while neural and kernel-based models tend to perform slightly lower.

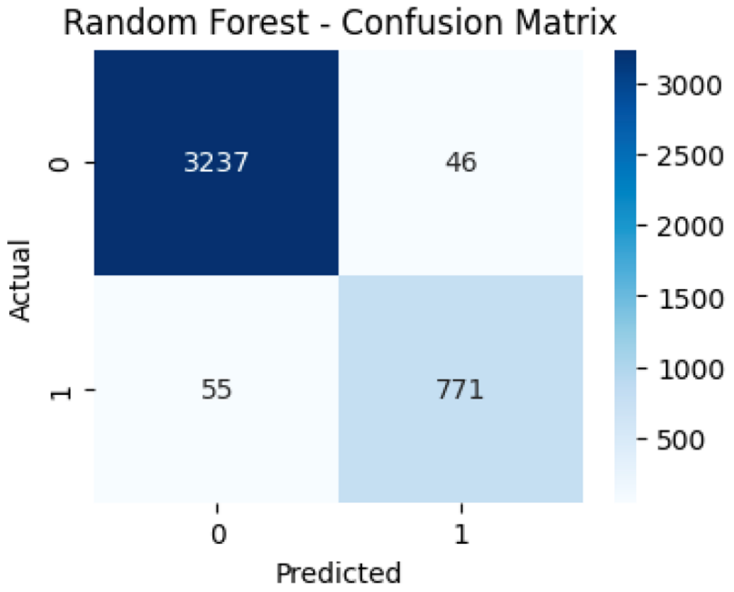

Fig. 2. Confusion matrix for the Random Forest model on the Bangladesh flood dataset, illustrating correct and incorrect classifications of flood and non-flood events

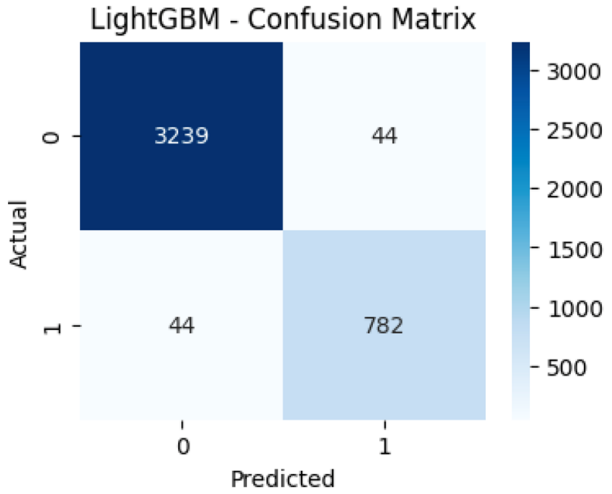

Fig. 3. Confusion matrix for the LightGBM model on the Bangladesh flood dataset, illustrating correct and incorrect classifications of flood and non-flood

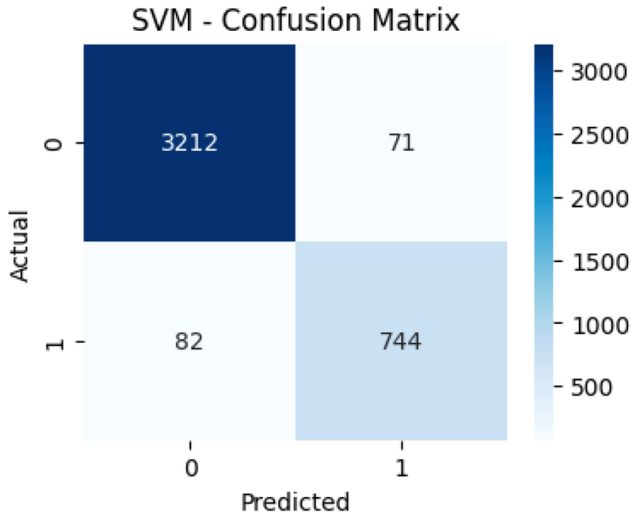

Fig. 4. Confusion matrix for the Support Vector Machine model on the Bangladesh flood dataset, illustrating correct and incorrect classifications of flood and non-flood events

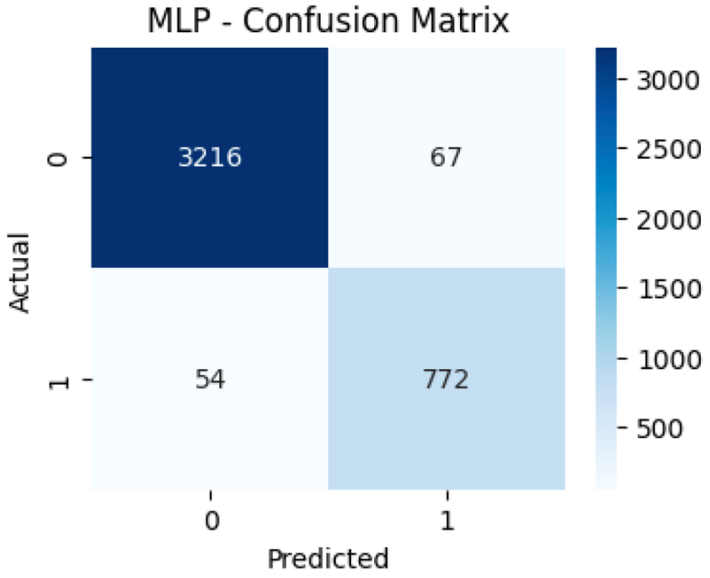

Fig. 5. Confusion matric for the Multilayer Perceptron model on the Bangladesh flood dataset, illustrating correct and incorrect classifications of flood and non-flood events

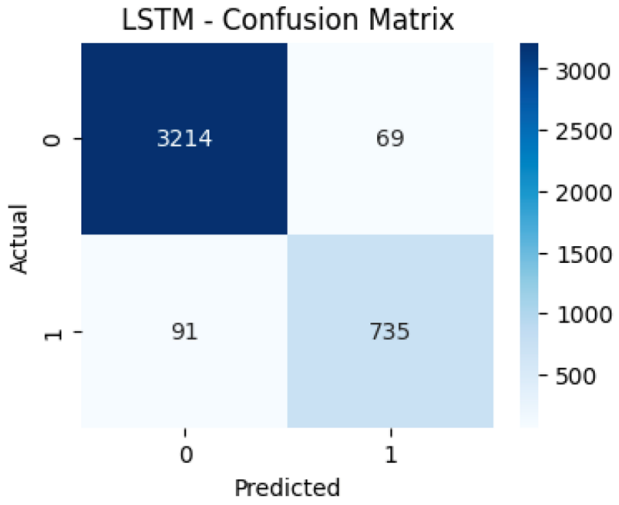

Fig. 6. Condusion matrix for the Long Short- Term Memory model on the Bangladesh flood dataset, illustrating correct and incorrect classifications of flood and non-flood events

Since the classes within the Bangladesh dataset were imbalanced, we have further evaluated the discriminative ability of all models using a Receiver Operating Characteristic (ROC) chart shown in Figure 7. The Receiver Operating Characteristic – Area Under the Curve (ROC-AUC) scores show that all models have performed well in separating between floods and non-flood events, with each model achieving a score above 0.98.

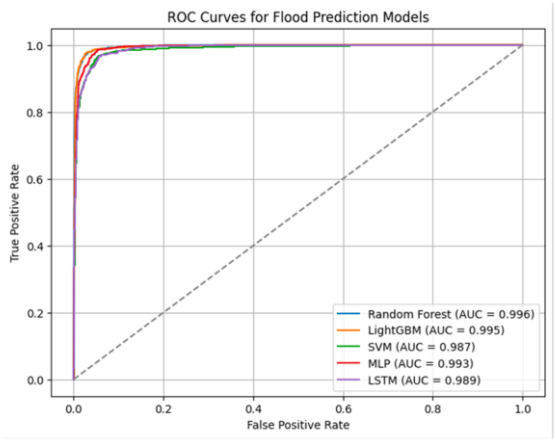

Fig. 7. ROC Curves for all flood prediction models evaluated on the Bangladesh dataset, illustrating each model's ability to distinguish between flood and non-flood events

Random Forest has achieved the highest AUC score of 0.996, followed by LightGBM with 0.995, MLP with 0.993, LSTM with 0.989, and lastly SVM showed a strong AUC score of 0.987 as well. These results confirm that despite the performance differences in other evaluation metrics such as precision and F1-score, all models have achieved near-perfect class separability when it comes to the AUC metric, with the Random Forest providing the highest performance.

Regarding further interpretability, how the features affected the LightGBM model's decision process was also studied. This was done by calculating the number of times that a feature was used to split the data in the classification process. From this method, it appeared that the most

important features in the model were the amount of rainfall, the station location, and the time of recording. The full ranking of feature importance is shown in Figure 8 below.

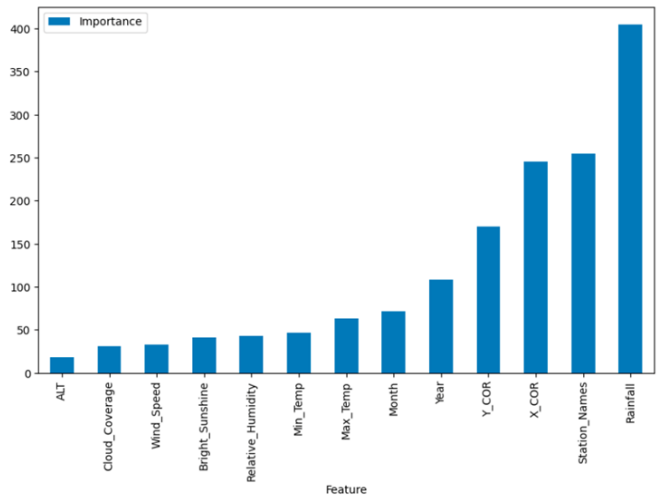

Fig 8. Feature importance ranking for the LightGBM model on the Bangladesh flood dataset, showing the relative contribution of each input variable to the model's decision-making

Overall, it appears that the LightGBM model performed the best for a variety of reasons. First, this model performed well because it handles categorical and numerical features with ease. Second, it has built-in techniques to prevent overfitting which allows the model to generalize well. Lastly, flooding is influenced by multiple interacting variables and LightGBM can handle these interactions automatically. Moreover, Random Forest performed well since it handles noise, outliers, and nonlinearities. Finally, the neural models such as LSTM did not perform as well because the Bangladesh dataset is not connected in a continuous time sequence but rather in a time sequence for each recording station. This also could explain why the LSTM model worked better for the Minnesota data rather than the LightGBM model.

In general, the results compiled in this paper showed that the gradient boosting-based approach, specifically the LightGBM model, provided superior performance and generalization across all evaluated metrics. For flood prediction, LightGBM can be considered an efficient and reliable model within the scope of this research. In addition, LSTM can also be used if the data were sequential.

5. FURTHER RESEARCH

There are several directions for further research on this project. First, given that the Minnesota River data only covered river height, more data from Minnesota, such as weather and other recording points, should be collected. This could improve the generalizability of the models because Minnesota and Bangladesh have different climates. In addition, other inputs such as satellite data, soil moisture, and topographic variables could be incorporated in the models. Second, given the high performance metrics, the models could be reviewed for potential data leakage or overfitting. This could be done by splitting the data before applying data transformations or by performing cross-validation. Additionally, the statistical rigor of the project could be improved through confidence intervals or repeated experiments. Beyond individual architectures, other models could be developed through hybridization. For example, a Convolutional Neural Network (CNN) could be used for its effectiveness with spatial features and then combined with the LSTM to learn temporal patterns to potentially develop a stronger

prediction model. For spatial features, models can be developed with Geographic Information Systems (GIS) to analyze flood data geographically using elevation, water flow, and spatial patterns. Finally, a real-time forecasting model could be created with continuously updating weather data to provide an early warning system for floods.

6. CONCLUSION

This research compares the performance of several machine learning models for flood prediction using two distinct datasets: a meteorological dataset from Bangladesh and a time-series river-height dataset from Minnesota. For the Bangladesh dataset, LightGBM has provided the highest overall performance across the accuracy, precision, recall, F1-score, NSE, and KGE. However, Random Forest has also performed competitively and achieved the highest ROC-AUC score, showing a strong class separability. Other evaluated models have also achieved high scores but were slightly less effective in comparison to LightGBM and Random Forest.

For the Minnesota time-series dataset, LSTM achieved the highest accuracy and performed better than other models due to its capability in learning the time-dependent patterns. This shows that the type of data has a strong influence on model performance. Boosting models tend to perform well with tabular data, while neural networks such as LSTM perform better with time-series data. The study notes the potential of machine learning methods in flood forecasting systems and shows the importance of selecting the machine learning model that aligns best with the structure of the data within that system, as well as the importance of certain features and their influence on flood prediction. Future improvements may be achieved by including more hydrological variables, incorporating satellite and GIS-based spatial features, and developing hybrid deep-learning architectures for more accurate real-time flood prediction.

REFERENCES

[1] Ghorpade et al. (2021, ICSCC) – Flood Forecasting Using Machine Learning: A Review
[2] Nevo et al. (2022, Google Research operational framework) – Flood forecasting with ML models in an operational framework
[3] Mosavi et al. (2018, MDPI Water) – Flood Prediction Using Machine Learning Models: Literature Review
[4] Syeed et al. (2022, IEEE HORA) – Flood Prediction Using Machine Learning Models
[5] Nevo et al. (2020, arXiv) – ML-based Flood Forecasting: Advances in Scale, Accuracy and Reach
[6] Soares et al. (2025, Journal of Hydrology) – ML4FF: A machine-learning framework for flash flood forecasting
[7] Akinsoji, A., Li, Z., & Adepoju, A. (2024). Integrating machine learning models with comprehensive data strategies and optimization techniques to enhance flood prediction accuracy: A review. Journal of Hydrology and Environmental Systems
[8] Maspo, F., Khosravi, K., & Mosavi, A. (2020). Evaluation of machine learning approaches in flood prediction scenarios and their input parameters: A systematic review. Journal of Hydrology, 590, 125–141
[9] Kunverji, R., Sharma, S., & Patel, J. (2021). A flood prediction system developed using various machine learning algorithms. International Journal of Computer Applications, 183(5), 1–6