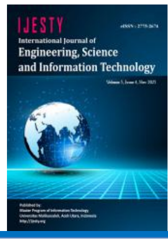
International Journal of Engineering, Science and Information Technology
Volume 5, No. 4 (2025) pp. 527-537
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i4.1179
Research Paper, Short Communication, Review, Technical Paper



Performance Evaluation of Machine Learning and Deep Learning for Rainfall Forecasting

Arief Andy Soebroto[1]*, Lily M. Limantara[2], Wayan Firdaus Mahmudy[1], Moh. Sholichin[2], Nurul Hidayat[1], Agi Putra Kharisma[1]

[1]*Department of Informatics, Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia*
[2]*Department of Water Resources Engineering, Faculty of Engineering, Universitas Brawijaya, Malang, Indonesia*

**Corresponding author Email: ariefas@ub.ac.id*

The manuscript was received on 1 March 2025, revised on 10 July 2025, and accepted on 30 October 2025, date of publication 14 November 2025

Abstract

Climate change is a significant challenge for both humans and the environment, with its impacts increasingly felt across various regions of the world. The most evident consequence is the alteration of extreme weather patterns, which often lead to destructive and life-threatening natural disasters. Among these, extreme rainfall was the most damaging factor, frequently triggering floods. However, the increasing occurrence of related events outlined the urgent need for developing more accurate rainfall forecasting systems as a strategic measure for disaster risk reduction. This research adopted daily rainfall data from Samarinda City, collected between 2004 and 2012, to conduct prediction using both machine and deep learning methods. The implementation of machine learning methods, such as Support Vector Regression (SVR), enabled the model to learn from historical data and uncover complex patterns, resulting in accurate forecasts and improved adaptability to climate variability. Meanwhile, deep learning models, including Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM), enhanced prediction performance by capturing more intricate and abstract data relationships. Performance evaluations conducted using Mean Absolute Error (MAE) and Mean Squared Error (MSE) showed that deep learning outperformed machine learning in accuracy. The LSTM model achieved the best performance, with loss values of 0.0482 and 0.0527 for MSE and MAE, respectively. The advantage of deep learning lies in its ability to build more complex models for handling non-linear problems and to learn data representations at various levels of abstraction, which has led to more accurate results. Furthermore, LSTM surpassed RNN by effectively overcoming the vanishing gradient issue, allowing for more stable and efficient training that led to superior predictive performance.

Keywords: *Climate Change, Deep Learning, Forecasting, Machine Learning, Rainfall.*

1. Introduction

Climate change is a major challenge for both humans and the environment, with its impacts increasingly felt across various parts of the world [1]. The most evident effect is the shift in extreme weather patterns, which can lead to destructive and deadly natural disasters [2]. Extreme rainfall is also damaging as it often triggers flood events [3][4]. In addition, flood has serious consequences such as damaging property and infrastructure, including exposing the affected communities to health and economic risks [5]. Regions frequently affected by climate change, such as Indonesia, have repeatedly experienced flood disasters [6], resulting in the need for risk reduction and adaptation planning to address this threat [7].

Based on the description above, forecasting plays a significant role in natural disaster risk reduction [8]. Traditional methods associated with weather forecasting, namely statistical methods, provide valuable contributions, although these inadequately address the complexity and uncertainty related to climate change [9]. Accurate rainfall forecasting enables authorities to design flood reduction measures more effectively, including early warning systems and emergency response plans [10]. The integration and use of real-time weather data and advanced forecasting models helped improve forecasting accuracy, enabling rapid response to flood threats. This also prompted clear communication of related risks to the public, which was essential in terms of ensuring safety and adequate preparedness [11].

In this context, artificial intelligence (AI) has advanced significantly, particularly in machine learning and deep learning methods, which have shown promising breakthroughs in rainfall forecasting. Machine learning is a branch of AI that uses algorithms and statistical models to identify patterns and make predictions from data [12]. The use of related methods such as Support Vector Regression (SVR) for rainfall forecasting enables models to learn from historical data, detecting complex patterns and providing more accurate predictions, including better adaptation to climate change [13]. Meanwhile, deep learning is a subfield of machine learning that uses complex artificial neural network architectures to learn hierarchical data representations [14]. The adoption of its models, namely Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM), enhanced rainfall prediction performance. This was realised by allowing the models to understand more complex and abstract data relationships [15]. The application of deep learning in rainfall forecasting focuses on its ability to automatically extract important features from weather data, enabling the model to learn complex and abstract patterns without manual specification [16]. The integration process allows for more accurate and adaptive modelling in response to complex climate pattern changes [17]. Therefore, both deep and machine learning need to be tested to obtain accurate evaluation results for each method.

Testing plays a crucial role in outlining the advantages of Mean Squared Error (MSE) and Mean Absolute Error (MAE) in rainfall prediction. Furthermore, MSE captures the squared differences between predicted and actual observations, providing a clearer indication of the magnitude beneficial for identifying and addressing outliers or large deviations observed in the rainfall dataset [18]. The simpler method of MAE offers a more intuitive interpretation of average prediction error [19]. Its strength lies in the ability to provide a direct indication of how far the average prediction deviates from the actual value with less sensitivity to outliers [20]. This testing provides deeper insights into the performance evaluation of methods used in solving rainfall forecasting problems.

The daily rainfall data provided by the Meteorology, Climatology and Geophysics Agency (*Badan Meteorologi Klimatologi dan Geofisika*/BMKG) for Samarinda City served as a crucial basis for weather forecasting. In addition, the identification of seasonal patterns and trends enabled the understanding of the evolving characteristics associated with rainfall over time [21]. This prompted the application of machine and deep learning methods to generate accurate data modelling and forecasting of future rainfall [22]. The methods mainly focused on model validation by comparing predictions with actual data to ensure consistent and reliable performance. BMKG further provided invaluable support in enhancing weather forecasting capabilities, helping communities and other stakeholders make informed and sustainable decisions. This present research contributed significantly to efforts in reducing the risks of natural disasters caused by extreme rainfall, improving community preparedness and resilience to the impacts of global climate change.

This research outlined a gap in the literature regarding rainfall forecasting in Samarinda City. Currently, no in-depth research has addressed the use of both deep and machine learning methods in the context of rainfall forecasting. Based on this perspective, the gap was filled by investigating the effectiveness of both methods in forecasting rainfall in the Samarinda region. The research also made significant contributions to the development of more accurate and reliable forecasting methods to enhance the understanding and reduction of extreme weather risks in the city. Rainfall forecasting was conducted using machine learning and deep learning. The performance evaluation was used to determine the appropriate parameter settings for related cases in Samarinda, specifically when machine and deep learning models were trained.

2. Literature Review

AI is an interdisciplinary field that combines computer and cognitive sciences, regarded as a focal point in modern technological innovation [23]. In developing AI applications, various aspects, including algorithms, data processing and ethical implications, must be considered. The application process has led to significant changes across various industries, ranging from healthcare to finance, thereby driving the advancement of sophisticated technologies [24]. A deep understanding of the theoretical foundations of AI is essential to optimise its potential in promoting innovation and social progress.

Machine learning refers to a branch of AI that focuses on developing algorithms, enabling computers to learn from data, as well as make decisions or predictions without being explicitly programmed [25]. Prior research has described the integration of machine learning technology with respect to its ability to identify complex patterns and make predictions based on historical data [26]. This has transformed the way information is processed, including decisions and the solving of complex problems. The application of machine learning has found widespread use across various industries, namely e-commerce, banking, healthcare and transportation. Therefore, deeply understanding the theoretical basis of machine learning plays an essential role in exploiting the potential of this technology in enhancing efficiency, innovation and advancement across diverse science fields.

Deep learning, another branch of AI, focuses on machine learning through hierarchical data representation [27]. Previous research has concentrated on its integration due to the ability to address complex problems such as pattern recognition, natural language processing and computer vision. Meanwhile, deep learning extracted relevant features from data at high abstraction levels through the use of deep artificial neural network architectures [28]. The application process has led to significant advances in various fields such as industrial automation, drug development and medical data analysis [29]. A profound understanding of the theoretical basis of deep learning is essential in terms of exploiting its potential in addressing complex challenges across multiple scientific disciplines. Figure 1 is a diagrammatic representation of AI, including machine and deep learning concepts.

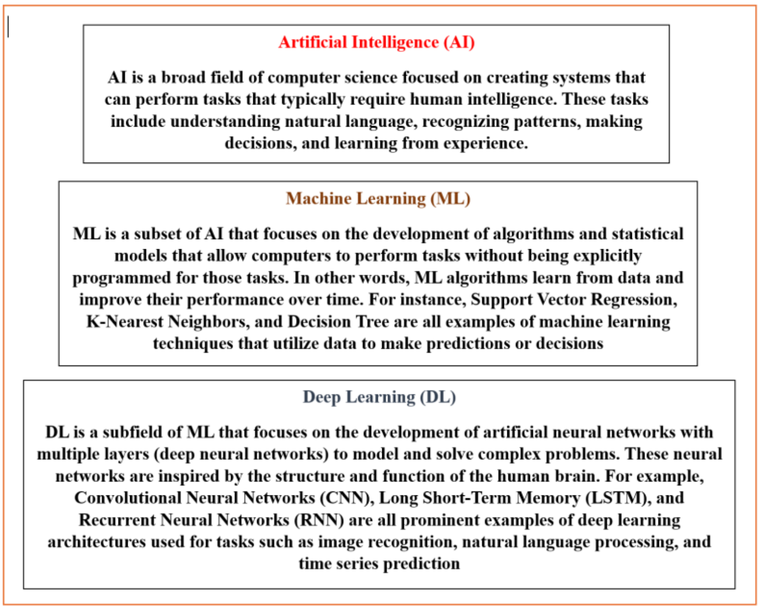


Fig 1. Concept of Artificial Intelligence, Machine Learning, and Deep Learning

Support Vector Regression (SVR) is an interesting and effective method in the context of rainfall forecasting. This method enabled the modelling of relationships between complex weather variables by considering the non-linear characteristics of rainfall data [30]. SVR further adjusted the optimal hyperplane to minimise prediction errors by identifying support vectors, regarded as significant points in the feature space [31]. The kernel functions used allowed effective handling of non-linear rainfall data [32]. The strength of SVR in managing outliers is another significant advantage, considering that rainfall data often contains extreme anomalies. However, carefully tuning the parameters enabled the achievement of optimal forecasting performance. This included considering the computational complexity, specifically when handling large datasets. SVR offered a powerful method for rainfall forecasting, particularly in situations where weather patterns are non-linear and the presence of outliers is significant.

Following the description above, RNN plays a promising role in rainfall forecasting. This method is particularly well-suited for handling sequential data associated with weather data, which can be viewed as a time series. The main advantage of RNN is the ability to consider previous context during the forecasting process, enabling the recognition of long-term patterns in rainfall data [33]. The non-linear activation functions also allowed for more complex modelling of the relationships between weather variables. In practice, RNNs tend to face issues such as gradient explosion and limited long-term memory, which affect forecasting performance specifically in cases where distant historical information must be considered [34]. However, proper architectural adjustments and technical issue management allowed RNN to remain a promising method in rainfall forecasting when handling complex sequential data.

LSTM is regarded as a highly effective method in rainfall forecasting. This type of RNN architecture is specifically designed to address long-term memory issues and overcome the vanishing gradient problem [35]. Furthermore, the ability to retain important historical information over long periods, including incorporating new inputs, made it suitable for forecasting rainfall influenced by past weather conditions. The method is also capable of understanding and capturing complex patterns in weather data, specifically due to the ability to model non-linear relationships between weather variables [36]. Its use requires careful parameter tuning and appropriate architectural design to achieve optimal forecasting performance [37]. In this context, LSTM represents an attractive and effective choice in rainfall forecasting, particularly in cases where long-term weather history and complex inter-variable relationships need to be considered.

Another relevant evaluation metric in rainfall forecasting analysis is MSE, responsible for measuring the average of the squared differences between predicted and actual observed values [38]. In addition, MSE is sensitive to discrepancies, implying that large errors contributed significantly to the overall value [39]. Its use offered a more comprehensive picture of forecast quality by outlining both the magnitude and direction of errors. MSE is commonly used in statistical modelling and forecasting to assess the accuracy of rainfall predictions, which offer valuable insights for decision-making in agriculture, water management and disaster risk reduction.

MAE is a commonly used evaluation metric in rainfall forecasting, considering that this metric measures the average of the absolute differences between predicted and actual observed values [40]. MAE also provided an overview of the extent the average rainfall prediction deviated from the actual value in millimetres. Its use offered information about the overall forecasting error level regardless of the direction [41]. A lower value represented a more accurate forecasting model during the prediction process [42]. MAE often served as a criterion for evaluating the performance of rainfall forecasting models and comparing the advantages among various forecasting methods. The metric played a crucial role in supporting decision-making related to disaster risk, agricultural planning and water resource management across multiple sectors.

3. Methods

3.1. Materials

The daily rainfall dataset in Samarinda City from 2004 to 2012 comprised various weather features such as minimum and maximum average temperature, average humidity, duration of sunlight, as well as maximum wind speed and direction, including its average. The target of this dataset was the amount of rainfall collected in a day. The process enabled the analysis and modelling to understand rainfall patterns, as well as the relationship with other weather variables. This included building predictive models and conducting trend analysis to estimate rainfall based on weather conditions, and understanding changes over the years. Table 1 shows the rainfall dataset of Samarinda City.

Table 1. Rainfall Dataset of Samarinda City

Tn	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg	ddd_car
25	35	27.7	82	2.7	2.6	4	360	0	N
24	33	26.9	89	14	2.9	4	135	1	N
24	29.6	26.1	90	9.2	0	4	90	0	N
24	31.8	27.7	85	0	2.7	3	45	1	N
25	33.8	27.2	82	0	0.8	3	90	0	N
23	32.6	27.8	82	33.5	3	3	360	1	N
23	32.6	27.1	80	35	3	3	90	0	N
23	29.4	26.2	84	0	1	2	135	0	N
24	34	28.1	78	0	7.8	5	45	2	N
24	34.2	28.8	74	0	8	6	45	2	N

3.2. Methods

The development of machine learning and deep learning for rainfall forecasting represented an effort to address the impact of floods effectively. The implementation process enabled a deeper analysis of rainfall patterns and characteristics, which served as the basis for decision-making in disaster reduction and responses. This research accurately forecasted rainfall activities, intensifying the response to disaster events. The analysis included the adoption of a comprehensive methodological framework for rainfall forecasting. In addition, the framework consisted of data collection, preprocessing, selection of machine and deep learning architectures, data training and testing, as well as results evaluation.

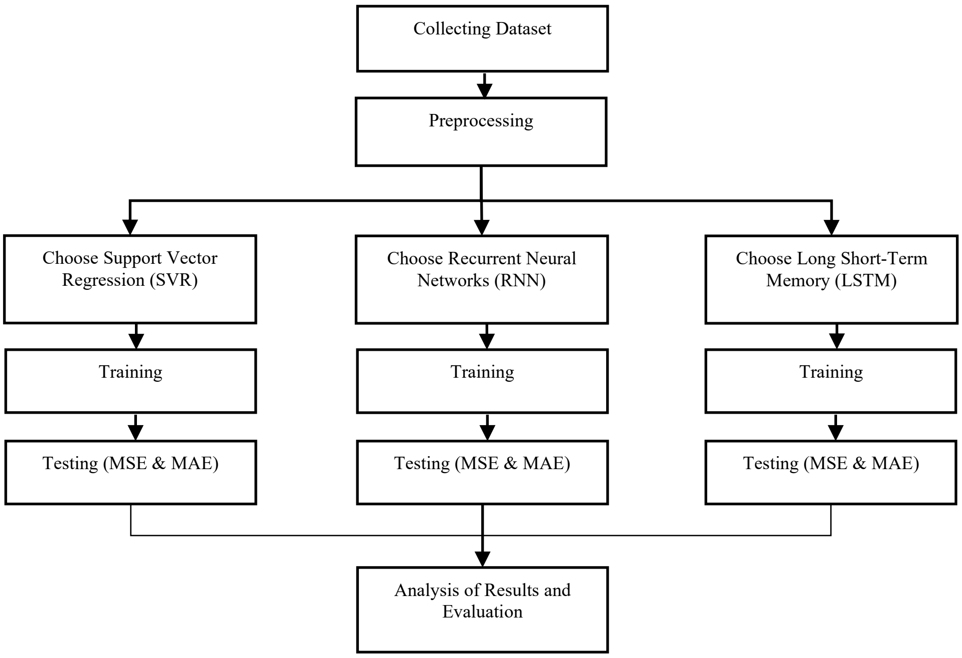


Fig 2. Research Flowchart

The research flowchart started with the daily collection of rainfall data from BMKG. The collected data was further subjected to preprocessing to ensure its readiness for algorithm training [43]. This included handling missing values and applying normalisation methods. Moreover, the commonly used forecasting methods were machine learning methods, namely SVR and deep learning architectures such as LSTM and RNN. After the selection of the appropriate machine and deep learning methods, the data was trained using these models by feeding the preprocessed features to generate accurate rainfall predictions [44]. This was followed by evaluating the models using unseen test data to assess forecasting performance. Evaluation metrics such as MAE and MSE were frequently used to measure the accuracy of the predictions [45][46]. Finally, the results must be thoroughly analysed to assess model performance and identify any existing limitations.

In this context, SVR was carried out by initialising predefined parameters such as epsilon, C, and the kernel. The next step included training the model using the relevant data. During the training phase, the model constructed the kernel matrix, solving an optimisation problem to determine the optimal alpha coefficients. Furthermore, SVR defined the selection of support vectors, calculation of model weights and bias, as well as storage of essential information. This included support vectors, weights, and bias for future use, with the entire SVR process shown in Figure 3.

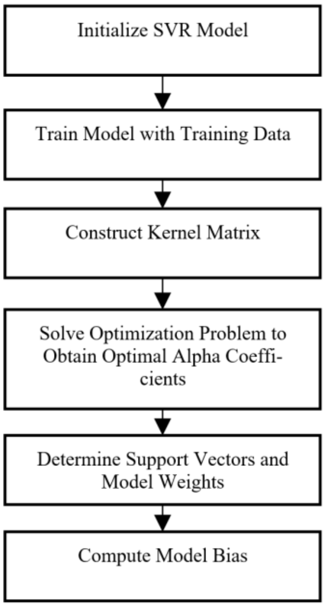


Fig 3. Support Vector Regression Flowchart

RNN was performed by initialising model parameters such as weights and biases. The first step entailed feeding training input data sequentially into the network. Furthermore, each timestep in the input sequence was subjected to an activation computation, where the input was multiplied by the corresponding weights. The result was added to the activation from the previous timestep and combined with a bias term. This computation produced an output, passed through an activation function such as sigmoid or tanh to generate a hidden state retaining relevant information from prior inputs. The diverse steps were repeated for each timestep in the training sequence, allowing the network to update its weights and learn temporal patterns associated with the data. Meanwhile, RNN served as the core of the learning process by enabling the network to incorporate and use past information for making predictions or understanding complex temporal structures [47]. This played an essential role in the journal context, describing the mathematical and conceptual processes underlying the construction of the RNN model. The processes included the provision of a clear understanding of the mechanism behind the effectiveness, as shown in Figure 4.

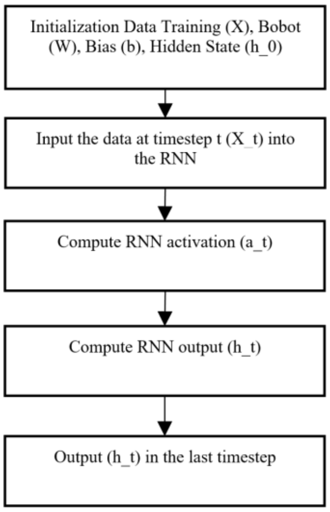


Fig 4. RNN Flowchart

LSTM networks started with the initialisation of weights and biases, a crucial step in preparing the network for learning. The input data, comprising a sequence of values, was fed into an LSTM network for processing. Additionally, the various gates, namely forget, input and output, responsible for regulating the flow of information in the memory cell, were adopted. Irrelevant data were discarded, and new information was input into the memory cell adhering to the decisions made by these gates [48]. The memory cell was updated following the new input, with the information retained or forgotten. The hidden state is updated in line with the output of the memory cell. The final result of the LSTM network was produced as the output, with the entire process shown in Figure 5.

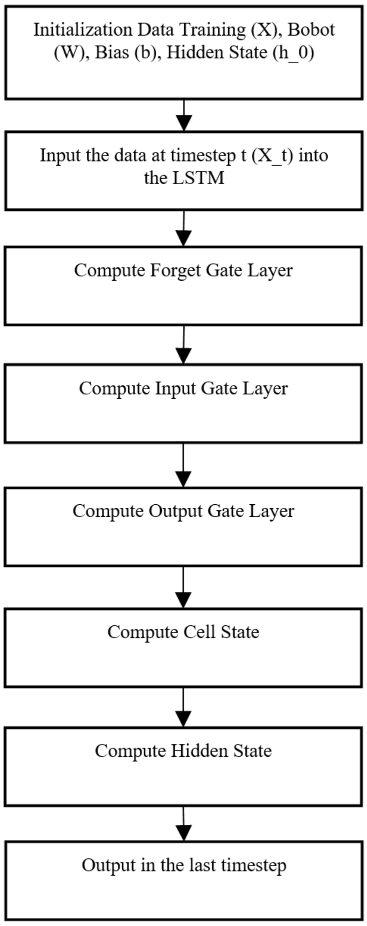


Fig 5. LSTM Flowchart

4. Results and Discussion

The performance evaluation of forecasting analysis was conducted to obtain the most optimal results for rainfall prediction. The evaluation process included the tuning of hyperparameters for each method; for example SVR method was tuned by adjusting the values of C and epsilon. The RNN method was tuned by adjusting the number of hidden units, layers, and learning rate. Additionally, the LSTM method was tuned using the same hyperparameters, including the number of hidden units, layers, and learning rate. The evaluation was performed by calculating the values of MAE and MSE.

The SVR method performed hyperparameter tuning by adjusting the values of C and epsilon. The C parameter was tested to apply a penalty for training errors, with a higher value imposing a larger penalty. The developed model selected a hyperplane with smaller training errors when using a higher C value, even if it led to a narrower margin. However, a lower C value allowed for a wider margin, and this permitted some data points to fall outside the margin, leading to higher training errors. SVR also tested the epsilon parameter to determine the width of the margin corridor around the hyperplane. A larger value allowed for more data points to fall in the corridor, and these could be beneficial with noisy information containing prediction uncertainty. Meanwhile, an excessively large epsilon led to an overly coarse or less precise model. Hyperparameter tuning graphs for the SVR method are shown in Figures 6 and 7.

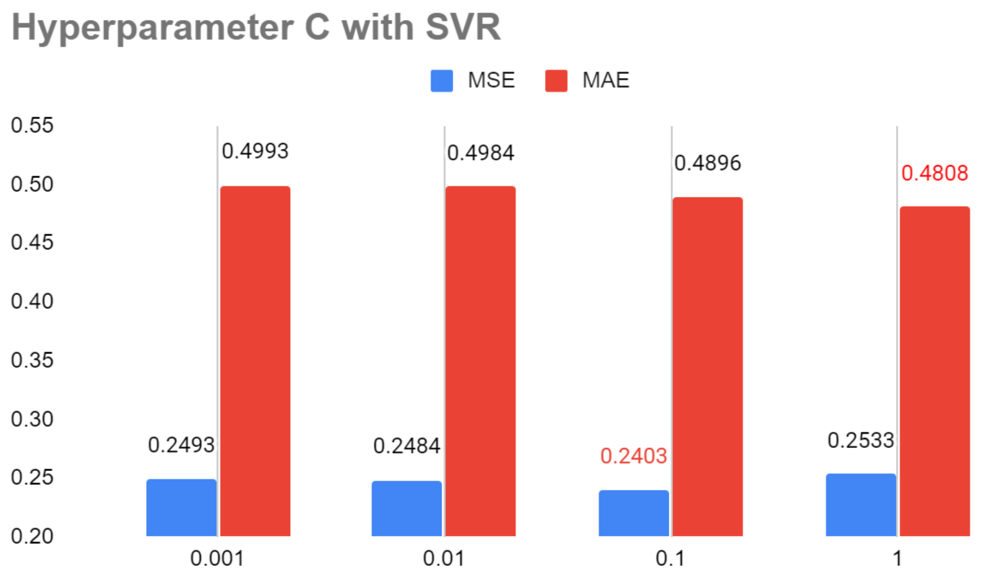


Fig 6. Hyperparameter C with SVR

Figure 6 shows the hyperparameter testing graph of the C parameter using the SVR method. The best performances were achieved with C values of 0.1 and 1 based on MSE and MAE evaluation. A value of 1 led to greater error in MSE evaluation because a higher C led to the sensitivity of the model to outliers, causing overfitting. However, a lesser value produced worse results than a C value of 1. This was because the algorithm became permissive of errors, causing underfitting due to insufficient model complexity.

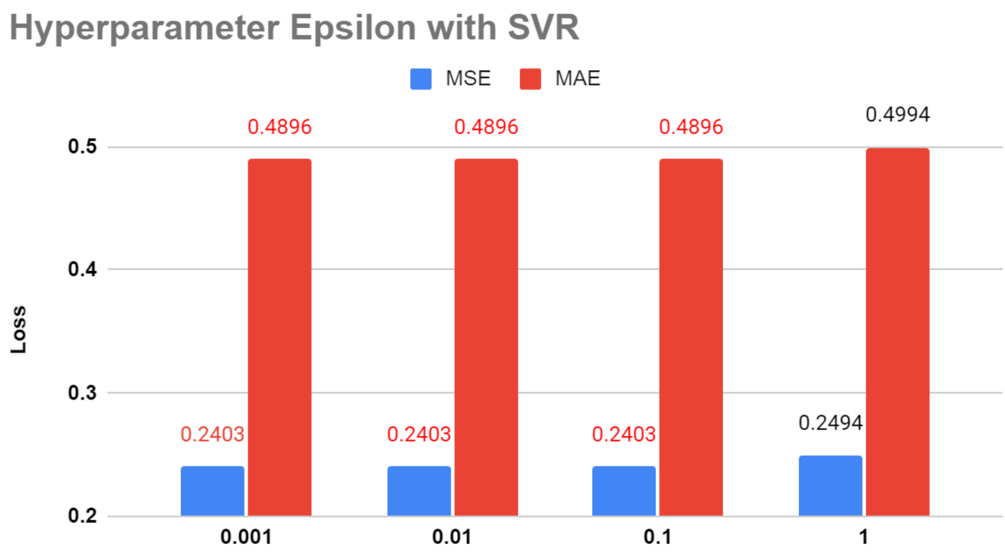


Fig 7. Hyperparameter Epsilon with SVR

The hyperparameter testing graph of the epsilon parameter using the SVR method is shown in Figure 7. The best performance was obtained with epsilon values of 0.1, 0.01, or 0.001, achieving a loss of 0.2403 and 0.4896 based on MAE and MSE evaluation, respectively. Testing with an epsilon value of 1 led to a higher error. This was because the model became permissive of training errors, leading to underfitting as a result of the inability to capture data complexity.

The RNN method performed hyperparameter tuning by adjusting the number of hidden neurons, layers, and learning rate. Different numbers of hidden neurons were tested to determine the optimal number of units for training. A higher number of neurons depicted that the model learnt more complex representations, while a lower number suggested its incapability of effectively learning from the data. The use of many hidden neurons led to overfitting. RNN also tested the number of layers to find the optimal model depth, and it was reported that a greater number enabled the model to learn more complex features, with fewer layers limiting the learning capacity. In this context, an excessive number of layers also increased the risk of overfitting. The learning rate was tested to control the step size taken when updating weights during training. An extremely large learning rate caused the model to overshoot local minima or diverge, resulting in unstable training. However, a learning rate that is too small caused the training to be slow and the model to get stuck in local minima or fail to converge. The results of hyperparameter tuning for the RNN method are shown in Figures 8, 9, and 10.

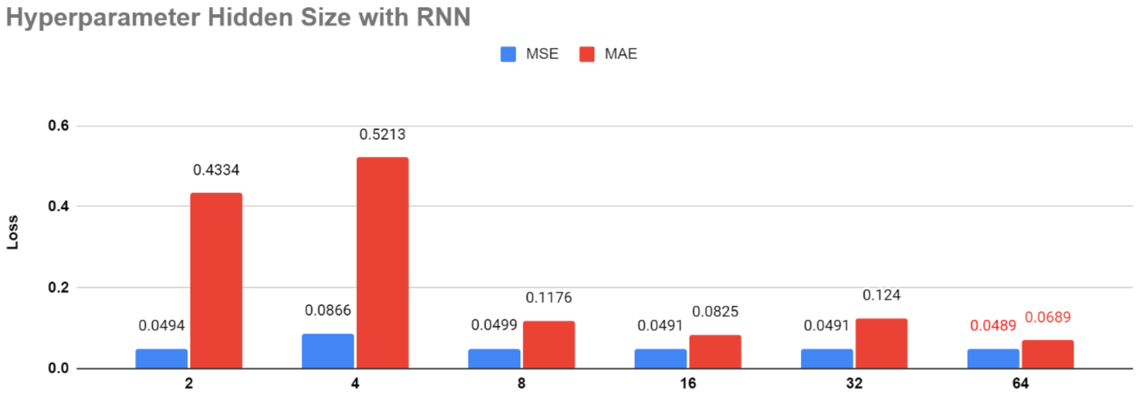


Fig 8. Hyperparameter Hidden Size with RNN

Figure 8 shows the hyperparameter testing graph of hidden size using the RNN method. The best performance was achieved with a hidden size of 64, resulting in a loss of 0.0689 and 0.0489 based on MAE and MSE evaluation, respectively. Testing with a smaller hidden size produced greater error values because the model experienced underfitting. This was due to the insufficient architectural complexity, which limited the ability to learn and represent the data effectively.

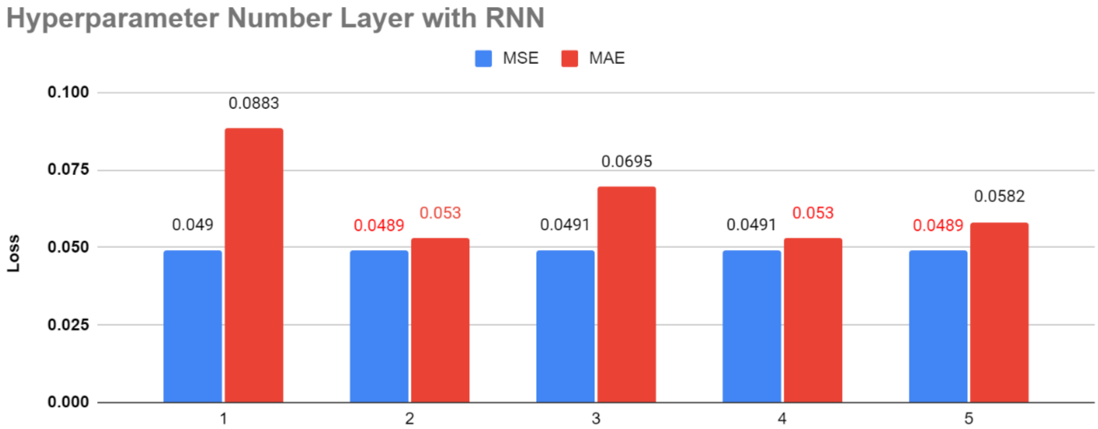


Fig 9. Hyperparameter Number Layer with RNN

The hyperparameter testing graph of the number of layers using the RNN method is shown in Figure 9. The best performance was obtained with two layers, causing a loss of 0.053 and 0.0489 based on MAE and MSE evaluation. The optimal number of layers was represented by achieving the lowest error values in both the MSE and MAE tests. Furthermore, the testing conducted on a higher number of layers produced greater error values because the model experienced overfitting. This was due to excessive architectural complexity, which caused the model to overlearn the data, as well as become overly sensitive.

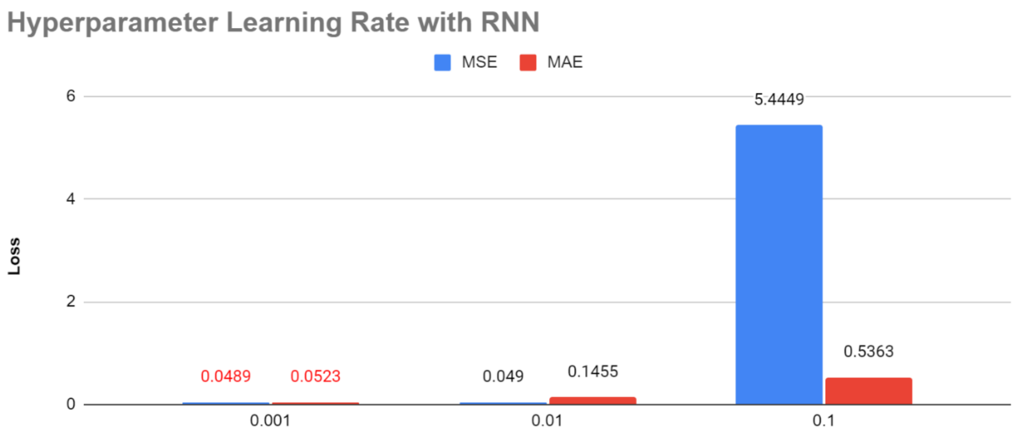


Fig 10. Hyperparameter Learning Rate with RNN

The hyperparameter testing graph of the learning rate using the RNN method is shown in Figure 10. The best performance was achieved with a learning rate of 0.001, resulting in a loss of 0.0523 and 0.0489 based on MAE and MSE evaluation, respectively. Additionally, testing with a higher learning rate produced greater error values because the model experienced overfitting. This was due to excessive architectural complexity, which caused the model to overinterpret the data, becoming overly sensitive.

The LSTM method performed hyperparameter tuning by adjusting the number of hidden neurons, layers, and learning rate. Different numbers of hidden neurons were tested to determine the optimal number of units for training. A higher number of hidden neurons showed that the model learnt more complex representations, while a lower number suggested the inability to effectively learn from the data. The use of many hidden neurons caused the model to overfit, and this prompted the testing of the layers to determine the optimal depth. A greater number of layers enabled the model to learn more complex features, with fewer layers limiting its capacity to capture patterns. Furthermore, an excessive number of layers led to overfitting, and the learning rate was tested to control the step size taken when updating weights during the training process. When the learning rate is high, it could cause the model to overshoot local minima or even diverge, resulting in unstable training. A learning rate that is low slows the training process significantly, causing the model to become trapped in local minima or fail to reach convergence. The results of hyperparameter tuning using the LSTM method are shown in Figures 11, 12, and 13.

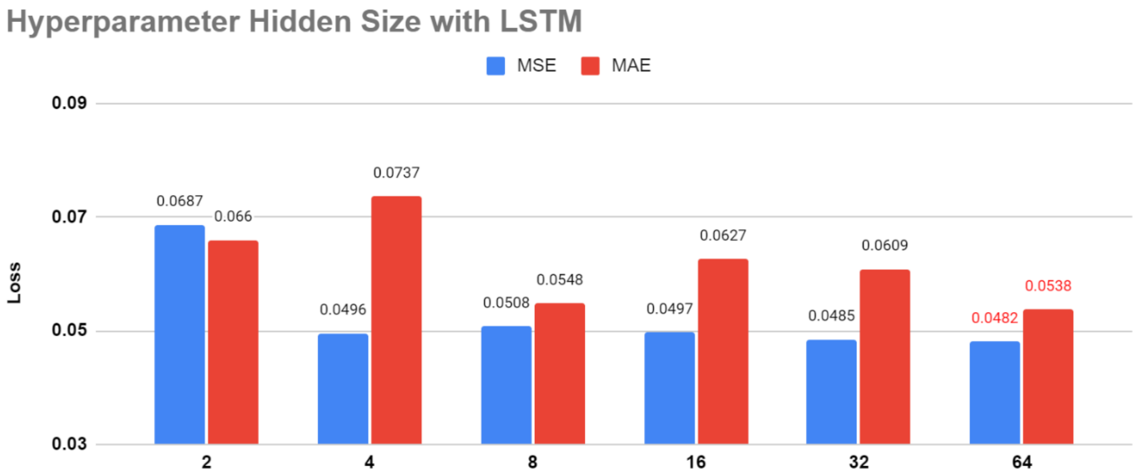


Fig 11. Hyperparameter Hidden Size with LSTM

Figure 11 shows the hyperparameter testing graph of hidden size using the LSTM method. The best performance was obtained with a hidden size of 64, causing a loss of 0.0538 and 0.0482 based on MAE and MSE evaluation. Meanwhile, testing with a smaller hidden size produced higher error values because the model experienced underfitting. This was due to an overly simple architecture that failed to capture the complex patterns in the data.

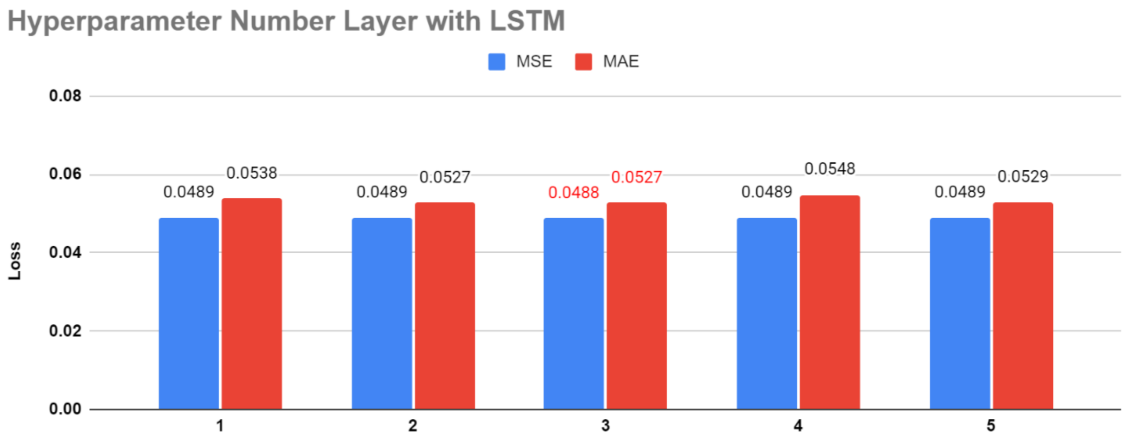


Fig 12. Hyperparameter Number Layer with LSTM

The hyperparameter testing graph of the number of layers using the LSTM method is shown in Figure 12. The best performance was achieved with three layers, resulting in the loss of 0.0527 and 0.0488 based on MAE and MSE evaluation, respectively. Furthermore, testing conducted on fewer than three layers produced higher error values because the model experienced underfitting. This was due to an overly simple piece of architecture that failed to capture the complexity of the data. Testing with more than three layers also amounted to higher error values because the model experienced overfitting due to excessive architectural complexity, leading to failure in data generalisation and overly sensitivity.

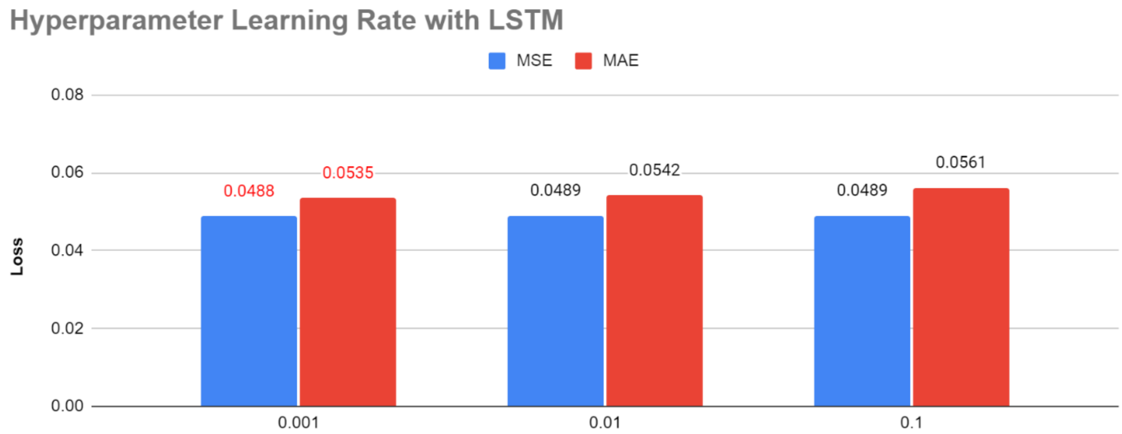


Fig 13. Hyperparameter Learning Rate with LSTM

Figure 13 shows the hyperparameter testing graph of the learning rate using the LSTM method. The best performance was achieved with a learning rate of 0.001, causing a loss of 0.0535 and 0.0488 based on MAE and MSE evaluation, respectively. Testing with higher learning rate values produced greater error because the model experienced overfitting due to excessive architectural complexity. In this context, the complexity caused the model to fail in generalising the data, thereby becoming overly sensitive.

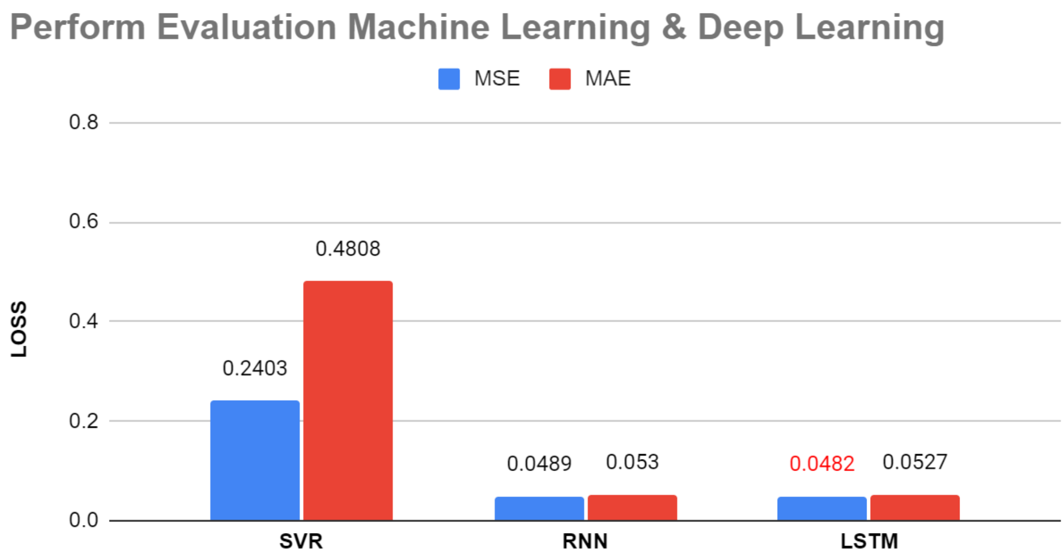


Figure 14. Perform Evaluation Machine Learning & Deep Learning

The performance evaluation graph of machine learning and deep learning methods is shown in Figure 14. The results proved that deep learning methods achieved more optimal performance compared to machine learning. The best deep learning algorithm was LSTM, with a loss of 0.0482 and 0.0527 based on MSE and MAE evaluation, respectively. The superiority of deep learning over machine learning is focused on its ability to construct more complex models for handling nonlinear problems. This included the capacity to learn data representations at various abstraction levels, resulting in more accurate models as evidenced by lower loss values. LSTM outperformed RNN due to the capability of resolving the vanishing gradient problem that commonly occurs in RNN. Additionally, by addressing the vanishing gradient issue, LSTM enabled more effective training, which led to more optimal testing results.

5. Conclusion

In conclusion, the results showed that deep learning outperformed machine learning in forecasting rainfall in Samarinda City. These also proved that among deep learning methods tested, LSTM produced the best performance compared to RNN. The evaluation showed that LSTM, as a deep learning model, achieved loss values of 0.0482 and 0.0527 based on MSE and MAE, respectively. Moreover, the best-performing LSTM model was obtained using 64 hidden units, 3 layers, and a learning rate of 0.001. The experimental results confirmed that these parameter settings led to the most optimal outcomes. Deep learning methods outperformed machine learning due to the ability to construct more complex models responsible for handling non-linear problems. This included the capacity to learn data representations at multiple levels of abstraction, resulting in more accurate models as reflected by lower loss values. Meanwhile, LSTM performed better than RNN because it resolved the vanishing gradient problem inherent in RNN, enabling stable and effective training processes.

Future research is advised not to use the RNN method, as it has been proven to perform worse than LSTM in forecasting rainfall in Samarinda City. Subsequent analyses may extend rainfall forecasting into related intensity classification. Furthermore, in this classification task, rainfall can be categorised into cloudy, light, moderate, heavy, very heavy, and extreme rain. Several studies were also motivated to explore other deep learning-based methods, such as Bidirectional Long Short-Term Memory (Bi-LSTM) or Gated Recurrent Unit (GRU), which have the potential to enhance the accuracy of rainfall forecasting and classification. This development is expected to provide more detailed and accurate information regarding weather conditions, allowing communities and stakeholders to adopt more appropriate actions based on the predicted level of rainfall intensity.

Acknowledgement

The authors are grateful to Badan Meteorologi Klimatologi dan Geofisika (BMKG) for providing the data that served as the primary foundation of this research. This research could not have been conducted effectively without access to high-quality data.

References

[1] O. Alizadeh, "Advances and challenges in climate modeling," Climatic Change, vol. 170, no. 1, p. 18, 2022. doi: 10.1007/s10584-021-03298-4.
[2] M. Bufalo, C. Ceci, and G. Orlando, “Addressing the financial impact of natural disasters in the era of climate change,” The North American Journal of Economics and Finance, vol. 73, p. 102152, Apr. 2024, doi: 10.1016/j.najef.2024.102152.
[3] F. Giglio, P. Frontera, A. Malara, and F. Armocida, "Materials and Climate Change: A Set of Indices as the Benchmark for Climate Vulnerability and Risk Assessment for Tangible Cultural Heritage in Europe," Sustainability, vol. 16, no. 5, p. 2067, 2024. doi: 10.3390/su16052067.
[4] C. Xie et al., “Advances in the study of natural disasters induced by the ‘23.7’ extreme rainfall event in North China,” Natural Hazards Research, Jan. 2025, doi: 10.1016/j.nhres.2025.01.003.
[5] M. L. Museru, R. Nazari, A. N. Giglou, K. Opare, and M. Karimi, “Advancing flood damage modeling for coastal Alabama residential properties: A multivariable machine learning approach,” The Science of the Total Environment, vol. 907, p. 167872, Oct. 2023, doi: 10.1016/j.scitotenv.2023.167872.
[6] A. Kurniadi, E. Weller, J. Salmond, and E. Aldrian, “Future projections of extreme rainfall events in Indonesia,” International Journal of Climatology, vol. 44, no. 1, pp. 160–182, Dec. 2023, doi: 10.1002/joc.8321.
[7] Md. Esraz-Ul-Zannat, A. Dedekorkut-Howes, and E. A. Morgan, “A review of nature-based infrastructures and their effectiveness for urban flood risk mitigation,” Wiley Interdisciplinary Reviews Climate Change, vol. 15, no. 5, May 2024, doi: 10.1002/wcc.889.
[8] K. M. A. Abdouli, A. Rai, G. Tenzin, U. Gayleg, N. Chhetri, and A. Chhetri, “Real-time flood forecasting in Amo Chhu using machine learning model and internet of things,” Cogent Engineering, vol. 11, no. 1, Jun. 2024, doi: 10.1080/23311916.2024.2370900.
[9] H. Xu, Y. Zhao, Z. Dajun, Y. Duan, and X. Xu, “Exploring the typhoon intensity forecasting through integrating AI weather forecasting with regional numerical weather model,” Npj Climate and Atmospheric Science, vol. 8, no. 1, Feb. 2025, doi: 10.1038/s41612-025-00926-z.
[10] A. A. Soebroto, L. M. Limantara, E. Suhartanto, and M. Sholichin, "Modelling of Group Decision Support Systems For Flood Alert With Nakayasu Synthetic Unit Hydrograph In Gadang Watershed," Journal of Southwest Jiaotong University, vol. 58, no. 1, 2023. doi: 10.35741/issn.0258-2724.58.1.13.
[11] A. A. Soebroto, L. M. Limantara, E. Suhartanto, and M. Sholichin, "Modelling of Flood Hazard Early Warning Group Decision Support System," Civil Engineering Journal, vol. 10, no. 2, pp. 614–627, 2024. doi: 10.28991/CEJ-2024-010-02-018.
[12] Z. Liu, N. Coleman, F. I. Patrascu, K. Yin, X. Li, and A. Mostafavi, “Artificial Intelligence for Flood Risk Management: A Comprehensive State-of-the-Art Review and Future Directions,” International Journal of Disaster Risk Reduction, vol. 117, p. 105110, Dec. 2024, doi: 10.1016/j.ijdrr.2024.105110.
[13] S. Q. Dotse, I. Larbi, A. M. Limantol, and L. C. De Silva, "A review of the application of hybrid machine learning models to improve rainfall prediction," Modeling Earth Systems and Environment, vol. 10, no. 1, pp. 19–44, 2024. doi: 10.1007/s40808-023-01835-x.
[14] M. M. Taye, "Understanding of machine learning with deep learning: architectures, workflow, applications and future directions," Computers, vol. 12, no. 5, p. 91, 2023. doi: 10.3390/computers12050091.
[15] S. S. Choudhary and S. K. Ghosh, "Analysis of rainfall and temperature using deep learning model," Theoretical and Applied Climatology, vol. 153, no. 1, pp. 755–770, 2023. doi: 10.1007/s00704-023-04493-2.
[16] O. A. Wani et al., “Predicting rainfall using machine learning, deep learning, and time series models across an altitudinal gradient in the North-Western Himalayas,” Scientific Reports, vol. 14, no. 1, Nov. 2024, doi: 10.1038/s41598-024-77687-x.
[17] W. Li, X. Gao, Z. Hao, and R. Sun, "Using deep learning for precipitation forecasting based on spatio-temporal information: a case study," Climate Dynamics, vol. 58, no. 1, pp. 443–457, 2022. doi: 10.1007/s00382-021-05916-4.
[18] P. K. Das, R. L. Sahu, and P. C. Swain, “Comparative analysis of machine learning models for rainfall prediction,” Journal of Atmospheric and Solar-Terrestrial Physics, vol. 264, p. 106340, Aug. 2024, doi: 10.1016/j.jastp.2024.106340.
[19] N. Haeruddin, E. Noersasongko, N. Purwanto, and N. Muljono, “A Multi-Model Framework for Rainfall Forecasting: Evaluating Performance Model Statistical, Machine Learning, and Deep Learning Methods,” 2025 International Conference on Smart Computing, IoT and Machine Learning (SIML), pp. 1–6, Jun. 2025, doi: 10.1109/siml65326.2025.11080798.
[20] X. Li, K. Li, S. Shen, and Y. Tian, "Exploring Time Series Models for Wind Speed Forecasting: A Comparative Analysis," Energies, vol. 16, no. 23, p. 7785, 2023. doi: 10.3390/en16237785.
[21] P. Mishra et al., “Modeling and forecasting rainfall patterns in India: a time series analysis with XGBoost algorithm,” Environmental Earth Sciences, vol. 83, no. 6, Mar. 2024, doi: 10.1007/s12665-024-11481-w.
[22] S. Chen et al., “Improving the accuracy of flood forecasting for Northeast China by the correction of global forecast rainfall based on deep learning,” Journal of Hydrology, vol. 640, p. 131733, Jul. 2024, doi: 10.1016/j.jhydrol.2024.131733.
[23] A. A. Alsumaiei, “Long-term rainfall forecasting in arid climates using artificial intelligence and statistical recurrent models,” Journal of Engineering Research, Mar. 2024, doi: 10.1016/j.jer.2024.03.001.
[24] N. Rane, S. Choudhary, and J. Rane, “Artificial intelligence for enhancing resilience,” Journal of Applied Artificial Intelligence, vol. 5, no. 2, pp. 1–33, Sep. 2024, doi: 10.48185/jaai.v5i2.1053.
[25] M. Goos and M. Savona, “The governance of artificial intelligence: Harnessing opportunities and mitigating challenges,” Research Policy, vol. 53, no. 3, p. 104928, Jan. 2024, doi: 10.1016/j.respol.2023.104928.

[26] T. Ahmad, R. Madonski, D. Zhang, C. Huang, and A. Mujeeb, "Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the context of smart grid paradigm," Renewable and Sustainable Energy Reviews, vol. 160, p. 112128, 2022. doi: 10.1016/j.rser.2022.112128.
[27] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," Archives of Computational Methods in Engineering, vol. 27, pp. 1071–1092, 2020. doi: 10.1007/s11831-019-09344-w.
[28] B. O. Akinsehinde, C. Shang, and Q. Shen, “Achieving reliable rainfall forecasting through ensemble deep learning, fuzzy systems, and advanced feature selection,” International Journal of General Systems, pp. 1–53, Mar. 2025, doi: 10.1080/03081079.2025.2471993.
[29] L. Herrmann and S. Kollmannsberger, “Deep learning in computational mechanics: a review,” Computational Mechanics, vol. 74, no. 2, pp. 281–331, Jan. 2024, doi: 10.1007/s00466-023-02434-4.
[30] M. Akbarian, B. Saghafian, and S. Golian, "Monthly streamflow forecasting by machine learning methods using dynamic weather prediction model outputs over Iran," Journal of Hydrology, vol. 620, p. 129480, 2023. doi: 10.1016/j.jhydrol.2023.129480.
[31] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Support vector machines and support vector regression," in Multivariate Statistical Machine Learning Methods for Genomic Prediction, Cham: Springer International Publishing, 2022, pp. 337–378. doi: 10.1007/978-3-030-89010-0_9.
[32] P. Das, D. A. Sachindra, and K. Chanda, "Machine learning-based rainfall forecasting with multiple non-linear feature selection algorithms," Water Resources Management, vol. 36, no. 15, pp. 6043–6071, 2022. doi: 10.1007/s11269-022-03341-8.
[33] S. F. Ahmed et al., "Deep learning modelling techniques: current progress, applications, advantages, and challenges," Artificial Intelligence Review, vol. 56, no. 11, pp. 13521–13617, 2023. doi: 10.1007/s10462-023-10466-8.
[34] M. Lin, D. Wu, S. Chen, J. Meng, W. Wang, and J. Wu, “Battery health prognosis based on sliding window sampling of charging curves and independently recurrent neural network,” IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1–9, Jan. 2024, doi: 10.1109/tim.2023.3348894.
[35] W. Li and T. Gao, “Water level prediction of firewater system based on improved hybrid LSTM algorithm,” IEEE Access, vol. 12, pp. 130305–130316, Jan. 2024, doi: 10.1109/access.2024.3444189.
[36] J. Jang, J. Han, and S. B. Leigh, "Prediction of heating energy consumption with operation pattern variables for non-residential buildings using LSTM networks," Energy and Buildings, vol. 255, p. 111647, 2022. doi: 10.1016/j.enbuild.2021.111647.
[37] P. Nguyen-Duc, H. D. Nguyen, Q.-H. Nguyen, T. Phan-Van, and H. Pham-Thanh, “Application of Long Short-Term Memory (LSTM) Network for seasonal prediction of monthly rainfall across Vietnam,” Earth Science Informatics, vol. 17, no. 5, pp. 3925–3944, Jul. 2024, doi: 10.1007/s12145-024-01414-3.
[38] J. Jian, S. He, W. Liu, S. Liu, and L. Guo, “A refined method for the simulation of catchment rainfall–runoff based on satellite–precipitation downscaling,” Journal of Hydrology, vol. 653, p. 132795, Feb. 2025, doi: 10.1016/j.jhydrol.2025.132795.
[39] Ö. A. Karaman, "Prediction of wind power with machine learning models," Applied Sciences, vol. 13, no. 20, p. 11455, 2023. doi: 10.3390/app132011455.
[40] M. Wang et al., “Estimating rainfall intensity based on surveillance audio and deep-learning,” Environmental Science and Ecotechnology, vol. 22, p. 100450, Jul. 2024, doi: 10.1016/j.ese.2024.100450.
[41] D. Trivedi, O. Sharma, and S. Pattnaik, “Minimization of forecast error using deep learning for real-time heavy rainfall events over Assam,” IEEE Geoscience and Remote Sensing Letters, vol. 21, pp. 1–4, Jan. 2024, doi: 10.1109/lgrs.2024.3378517.
[42] Z. Xu, B. Zhang, D. Li, W. S. Yip, and S. To, “Digital-twin-driven intelligent tracking error compensation of ultra-precision machining,” Mechanical Systems and Signal Processing, vol. 219, p. 111630, Jun. 2024, doi: 10.1016/j.ymssp.2024.111630.
[43] E. S. K. Tiu et al., "An evaluation of various data pre-processing techniques with machine learning models for water level prediction," Natural Hazards, vol. 110, no. 1, pp. 121–153, 2022. doi: 10.1007/s11069-021-04939-8.
[44] F. Jeribi et al., “A deep learning based expert framework for portfolio prediction and forecasting,” IEEE Access, vol. 12, pp. 103810–103829, Jan. 2024, doi: 10.1109/access.2024.3434528.
[45] A. Ariffien, S. Lamsir, R. Rasna, Q. Aini, and Moh. R. I. Matdoan, “Forecasting the inventory of milled dry grain using the lot sizing method at Markom Rice Mill,” International Journal of Engineering Science and Information Technology, vol. 5, no. 2, pp. 223–231, 2025. doi: 10.52088/ijesty.v5i2.817.
[46] S. Ardiansyah, R. K. Dinata, and A.-R. Ar-Razi, “Comparison of the results of the weighted moving Average method and the least absolute shrinkage and Selection operator method for predicting total palm oil production at PT. Mora Niaga Jaya,” International Journal of Engineering Science and Information Technology, vol. 5, no. 2, pp. 448–454, 2025. doi: 10.52088/ijesty.v5i2.862.
[47] F. Xie et al., "Deep learning for temporal data representation in electronic health records: A systematic review of challenges and methodologies," Journal of Biomedical Informatics, vol. 126, p. 103980, 2022. doi: 10.1016/j.jbi.2021.103980.
[48] A. Yunita et al., “Performance analysis of neural network architectures for time series forecasting: A comparative study of RNN, LSTM, GRU, and hybrid models,” MethodsX, vol. 15, p. 103462, Jul. 2025, doi: 10.1016/j.mex.2025.103462.