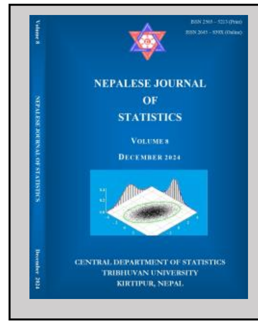
Rainfall Prediction using Long Short-Term Memory and Gated Recurrent Unit with Various Meteorological Parameters

Mahesh Pujara[1] and Nawaraj Paudel[2*]

Submitted: 12 September 2024; Accepted: 1 December 2024
Published online: 27 December 2024

DOI: https://doi.org/10.3126/njs.v8i1.73165



ABSTRACT

Background: Rainfall prediction is a critical task in meteorology and environmental science, with far-reaching implications for disaster preparedness, agriculture, and water resource management. Rainfall prediction can benefit greatly from the application of deep learning techniques like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, which have demonstrated great promise in time series forecasting.

Objective: To use LSTM and GRU to forecast rainfall in the Kathmandu metropolitan area using information gathered from the Department of Hydrology and Meteorology, Babarmahal, Kathmandu, Nepal.

Materials and Methods: Historical meteorological data was collected from Department of Hydrology and Meteorology, Babarmahal, Kathmandu, Nepal and preprocessed to create a suitable dataset. With this preprocessed dataset containing variables such as temperature, humidity, atmospheric pressure, wind speed, and direction, two deep learning methods, LSTM and GRU, were trained. To assess the performance, various evaluation metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and R^2 were used.

Results: The daily rainfall has been predicted using LSTM and GRU using 0.0001 learning rate, 50 epochs and 8 batch size. RMSE, MAE and R^2 values of LSTM are 2.51, 1.79 and 0.81 respectively. Similarly, RMSE, MAE and R^2 values of GRU are 2.31, 1.51 and 0.95 respectively.

Conclusion: Test results show that the GRU model's predictions are generally near to the actual recorded rainfall amounts, as evidenced by the fact that the model's test RMSE and MAE are fewer than those of the LSTM. A higher R^2 value of GRU suggests a better fit in the rainfall data, as more of the variance in the outcome is explained by the predictors.

Keywords: Gated recurrent unit, long short-term memory, mean absolute error, R^2, rainfall prediction.

Address correspondence to the author: Central Department of Computer Science and Information Technology, Institute of Science and Technology, Tribhuvan University[1, 2]
Email: maheshpujara007@gmail.com[1]; nawarajpaudel@cdcsit.edu.np[2*] (Corresponding author email)

INTRODUCTION

Because of the inherent complexity of meteorological systems and the fluctuation of atmospheric conditions, weather forecasting has always been a difficult undertaking. Rainfall, as a critical meteorological variable, is particularly challenging to predict accurately due to its spatial and temporal variability. Rainfall prediction is the task of estimating the amount and timing of rainfall in a given area or region using various data sources and prediction models. It involves analyzing historical rainfall patterns, as well as other environmental factors that affect rainfall, such as temperature, humidity, wind speed, and atmospheric pressure (Lenderink & Fowler, 2017). Predicting rainfall is an essential task in many domains, such as hydrology, agriculture, weather forecasting, and water resource management. Precise precipitation can assist farmers in crop planning, efficient water management, and readiness for impending floods or droughts. Furthermore, rainfall prediction can assist meteorologists in issuing timely and accurate weather alerts to the public, allowing individuals and communities to prepare for severe weather events (Aswin et al., 2018). Despite the importance of rainfall prediction, conventional statistical methods and numerical weather prediction models have shown limitations in capturing the intricate patterns and non-linear relationships in rainfall data. So, in recent years, machine learning techniques have become increasingly popular for rainfall prediction due to their ability to handle complex and high-dimensional data (Hassan et al., 2023). Specifically, deep learning models, such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network models, have shown promising results in predicting rainfall.

LSTM and GRU models are types of Recurrent Neural Network (RNN) that can handle sequential data and long-term dependencies. These models are designed to capture the temporal dependencies and patterns in the data, making them ideal for time series prediction tasks such as rainfall prediction. The manner in which LSTM and GRU models regulate the information flow inside the network is the main distinction between them. While GRU models employ update gates to regulate the flow of information, LSTM models use memory cells and gates to selectively forget or remember information (Zargar, 2021). Accurate rainfall prediction enables farmers to plan their crops better, make more informed decisions about irrigation and fertilization, and protect their crops against drought or flooding. Precise rainfall prediction helps in preparing for potential floods or landslides, and mitigating their impacts. It also helps in early warning and timely evacuation in the event of severe weather conditions. Accurate rainfall prediction assists in planning hydropower production, reducing the risk of power shortages, and ensuring a stable energy supply. Detailed prediction of rainfall enables meteorologists to issue timely and accurate weather advisories to the public, keeping people and communities safe during severe weather conditions.

In recent years, climate change has also become a concern, potentially influencing the rainfall pattern. Long-term monitoring and study of climate data are essential to understand and adapt to the changing rainfall patterns and their implications for agriculture, water resource management, and overall environmental sustainability. Because the environmental elements influencing rainfall are diverse and dynamic, accurately predicting rainfall is a difficult task. Since, traditional rainfall forecasting methods have limited accuracy, especially for short-term predictions, models such as LSTM and GRU have been shown to be promising for rainfall prediction. LSTM and GRU are RNN that are specifically designed to learn long-term dependencies in sequential data. This makes them well-suited for rainfall prediction, as rainfall data is often sequential and exhibits long-term dependencies. Therefore, this research paper focuses to use and optimize LSTM and GRU models to achieve accurate and reliable rainfall predictions, which can have significant implications for agriculture, water resource management, and disaster preparedness. In order to produce accurate and trustworthy rainfall predictions, the main goal of this research article is to employ and optimize LSTM and GRU models so that it can have a big impact on agriculture, water resource management, and disaster preparedness.

Both LSTM and GRU handle long-term dependencies in sequential data. While they share many similarities, they differ in terms of complexity and computational efficiency for a task like rainfall prediction. LSTM has complex architecture and higher parameter count whereas GRU has simpler architecture and fewer parameters. LSTM has slower training time, higher memory usage and suitability for long-term dependencies whereas GRU has faster training time, low memory usage and good for short- to medium-term dependencies. While LSTM and GRU models have shown promise in time series forecasting, specific research gaps for their application in rainfall prediction in Nepal include limited data availability, quality of data, regional variability in rainfall patterns, and geographic complexity. This study's primary goal is to forecast rainfall using LSTM and GRU by taking into consideration multiple meteorological parameters, such as air pressure, relative humidity, maximum and minimum temperatures, wind direction, and wind speed. The performance of these models for the rainfall prediction job is also compared and evaluated in this study using a dataset with six parameters that was received from Department of Hydrology and Meteorology, Babarmahal, Kathmandu, Nepal.

Literature review

Rainfall is a pivotal component of Earth's hydrological cycle, exerting profound impacts on agriculture, ecosystems, and human activities. Its intricate system is shaped by a multitude of factors. Rainfall is influenced by a complex interplay of meteorological factors that collectively determine the atmospheric conditions conducive to precipitation. One primary factor is temperature, as variations in air temperature contribute to the formation of low-pressure systems. When warm, moist air rises and cools, condensation occurs, leading to the development of clouds and eventually precipitation. Wind patterns play a crucial role in redistributing moisture across the atmosphere, with global wind systems like the Trade Winds and Westerlies guiding the movement of air masses. Other meteorological factors like atmospheric pressure and relative humidity also plays a significant role in rainfall. Numerous experiments have been carried out over time to forecast rainfall using different machine learning methods. In particular, LSTM and GRU are among the most widely utilized and efficient methods. LSTM and GRU are two types of RNNs that can handle long-term dependencies and sequential data. Since its introduction by the authors in (Hochreiter & Schmidhuber, 1997), LSTM has found extensive use in a wide range of fields, such as time series prediction, speech recognition, and natural language processing. A long-term information storage memory cell and three gates (an input gate, an output gate, and a forget gate) that control the information flow into and out of the cell are features of the LSTM. Authors in (Cho et al., 2014) presented GRU as an easier substitute for LSTM. GRU has two gates, update gate and reset gate, that regulate the flow of information. In some circumstances, GRU can be quicker and more effective than LSTM.

Authors in (Surta et al., 2023) evaluated performance of LSTM and GRU models for rainfall prediction in Palembang City using weather element data for ten years. This study concluded that LSTM model outperformed the GRU model. The RMSE of GRU model was 9.33 and R^2 was 0.54, while the RMSE of LSTM model was 7.45 and R^2 was 0.70. Authors in (Aswin et al., 2018) predicted the rainfall by using LSTM and ConvNet using the dataset obtained from the Global Precipitation Climatology Project. The study showed that ConvNet outperformed LSTM with RMSE of 2.44. In this study, the RMSE of LSTM was 2.55. In (Chhetri et al., 2020), authors used data gathered from Bhutan's meteorology department between 1997 and 2017 to assess the effectiveness of LSTM, GRU and BLSTM for rainfall prediction over the Simtokha region in Thimphu, Bhutan. Their study shows that LSTM slightly outperformed GRU and BLSTM with MSE of 0.0128 and RMSE of 0.113 where the MSE and RMSE of GRU and BLSTM are 0.129, 0.017, 0.138, 0.019, respectively. Authors in (Siami-Namini et al., 2018) compared compared the accuracy of ARIMA and LSTM for forecasting time series data. When these two methods were used on a set of financial data, the outcomes showed that LSTM outperformed ARIMA. To be more precise, the LSTM-based algorithm outperformed ARIMA in terms of prediction accuracy by an average of 85%, with an RMSE of 64.213 against 511.481 for ARIMA.

Authors in (Endalie et al., 2022)proposed a LSTM based prediction model to predict rainfall for Jimma, a region located in south-western Oromia, Ethiopia. The proposed model outperformed MLP, KNN, SVM, and DT with a RMSE of 0.01. Abdullahi and others examined the performance of the traditional LSTM and GRU models for forecasting streamflow on the ten MOPEX river basins, and they discovered that GRU performed better than LSTM with an RMSE of 1.41842 compared to LSTM's RMSE of 4.6526 (Muhammad et al., 2019). Authors in (Hess & Boers, 2022) used U-Net-based deep neural network to facilitate the learning of severe rainfall occurrences and suggested a frequency-based weighting of the loss function. Depending on the event size, the model increases the forecast skill of heavy rainfall events. It also produced a significantly more accurate representation of relative rainfall frequencies.

MATERIALS AND METHODS

The research is carried out in different phases; data collection, model implementation and performance evaluation. The model implementation involves implementing LSTM and GRU model for rainfall prediction. Implementation process of the full research process in detail can be viewed from the following figure.

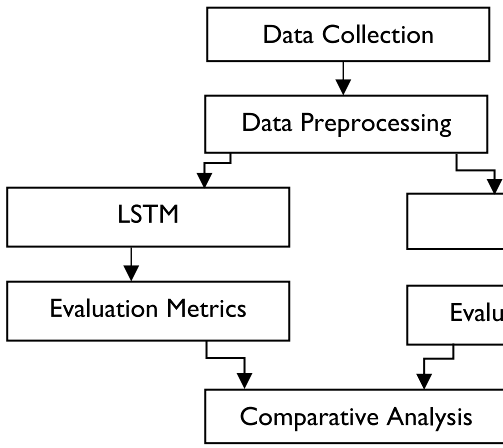


Fig. 1. Methodology overview for rainfall prediction using LSTM and GRU.

Data collection

Dataset for rainfall prediction is collected from the Department of Hydrology and Meteorology, Babarmahal, Kathmandu, Nepal. Different types of parameters were collected from the Kathmandu valley station "Tribhuvan International Airport" from the year 2015 to 2023 having 3100 instances. Six attributes make up the dataset: wind direction, wind speed, relative humidity, rainfall, maximum and minimum temperatures.

Data preprocessing

The data set contains several missing values that can be handled using imputation. Imputation is a technique that involves filling in missing values with estimated or predicted values. Time-series interpolation is one kind of imputation technique that take account the order and sequence of the data points, which can be valuable when dealing with time-series data like rainfall. There are several time-series interpolation techniques like linear interpolation, spline interpolation, or time-based interpolation (Géron, 2017). This research used linear interpolation to create new data points.

$$Linear\ interpolation\ (y) = y_1 + (x - x_1)\frac{y_2 - y_1}{x_2 - x_1}$$

where x_1 and y_1 are the succeeding coordinates; x_2 and y_2 are the preceding coordinates; x is the point to perform the interpolation; y is the interpolated value.

The data was then normalized using the min-max scaling technique. Scaling was done to make sure the model trains well, converges quickly, and yields findings that are understandable and relevant. Scaling is a preprocessing technique commonly used in machine learning to standardize or normalize the features of a dataset (Géron, 2017). The goal of scaling is to bring all features to a similar range or distribution without distorting the relationships between them. In this dissertation work scaling is particularly important because meteorological factors have different ranges and units. For example, temperature is in Celsius, while rainfall is in millimeters. These varying scales might affect the model's ability to learn effectively from different features, and scaling helps in mitigating that issue. Using techniques like Min-Max scaling normalizes the input features to a common scale, typically between 0 and 1 or with mean of 0 and standard deviation of 1. This normalization assists the model in learning patterns across different features more uniformly and efficiently, contributing to better model performance. Min-Max scaling preserves the relationships between data points while ensuring that all values fall within the same interval.

Formula for using min-max scaling:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here, X is the original cell value; X_{min} is the minimum value of the cell; X_{max} is the maximum value of the cell, and X_{sc} is the scaled value.

Model description

In the context of rainfall prediction, LSTM and GRU models capture the temporal dependencies in meteorological data. These models take into account historical rainfall patterns, temperature, humidity, wind speed, and other features to make predictions about future rainfall. In this research, LSTM layers are utilized to capture long-term dependencies in the sequential data. T55he input sequence is fed into the LSTM layer, and the hidden states and memory cells are updated at each time step. The output of the LSTM layer is passed through one or more dense layers for final rainfall prediction. Similar to LSTM, GRU layers are used to capture temporal patterns, but with a simpler architecture. GRU has an update gate that determines how much of the past information to keep and how much of the new information to incorporate. The GRU output is then fed into dense layers for rainfall prediction. The recurrent nature of LSTM and GRU allows the model to learn and capture temporal dependencies within the data. The memory cells and gating mechanisms enable the network to remember important past information and selectively update based on new observations. The internal mechanisms of LSTM and GRU models automatically learn relevant features from the input data. The models can identify patterns and relationships in the meteorological attributes that contribute to rainfall. The final output layer of the model produces the predicted rainfall values. The model is trained to minimize the difference between predicted and actual rainfall, optimizing its ability to generalize to unseen data.

Dataset, input features, target variable, and sequence length are given as inputs and sequences and target values are generated for training the LSTM and GRU model. It does so by iterating through the dataset, extracting sequences of data, and associating each sequence with its target value. A Sequential model with two LSTM and GRU layer followed by a dense output layer is defined separately. The model is compiled with the 'adam' optimizer and uses mean squared error as the loss function. A Sequential model is a linear stack of layers, and enables us to add layers one by one in sequence. The number of LSTM and GRU units (also known as neurons or cells) in the layer is also specified. These units act as memory cells that can store and update information over sequences. The activation function used in the LSTM cells is Rectified Linear Unit (ReLU). Then after the input dimension of LSTM and GRU model is specified. After the LSTM and GRU layer, a Dense layer is added. This layer is fully connected, which means each neuron in this layer is connected to every neuron in the previous layer. Finally, the models are compiled by using the compile method that configures the training process for the models. It requires two main arguments: optimizer='adam' that specifies the optimization algorithm to be used during training and loss='mean squared_error' that specifies the loss function to be minimized during training. Using a random search strategy, we identify the optimal collection of hyper-parameters for each model as shown in Table 1 below.

Table 1. Hyper parameters used.

Hyper-parameter	LSTM	GRU
Dropout	0.2	0.2
Loss function	Mean squared error	Mean squared error
Learning rate	0.0001	0.0001
No of epochs	50	50
Batch size	8	8
Optimizer	Adam $\beta 1=0.9$, $\beta 2=0.999$	Adam $\beta 1=0.9$, $\beta 2=0.999$

A dropout rate of 0.2 means that during training, randomly selected neurons in the LSTM or GRU layers will be "dropped out" or ignored with a probability of 20%. This helps in preventing the model from relying too much on specific neurons and promotes more robust generalization to unseen data. The difference between the target values and the model's predictions is measured by the loss function. For regression applications, such as rainfall prediction, where the objective is to minimize the squared differences between predicted and actual values, mean square error (MSE) is frequently selected. A small learning rate, such as 0.0001, is chosen to ensure slow and steady convergence during training. This helps the model avoid overshooting the optimal weights and improves stability. Training for 50 epochs means the model goes through the entire dataset 50 times. This value is chosen based on experimentation and monitoring the model's performance on a validation set. It ensures sufficient training without risking overfitting. A batch size of 8 indicates that the model updates its weights after processing 8 samples. The batch size is chosen small since the dataset contains only 3100 instances and the model performed best on small batch size value. The β1 and β2 parameters control the exponential decay rates for the first and second-order moments of the gradients. A commonly used configuration is β1 = 0.9 and β2 = 0.999, providing good performance in practice. This adaptive optimizer helps overcome challenges like vanishing or exploding gradients, often encountered in RNNs.

Dataset splitting

The data is split into training, validation, and testing sets at a ratio of 60:20:20. The majority of the data is allocated to training to ensure the model has enough data to learn complex patterns, especially for time-series tasks like rainfall prediction. A 20% validation set provides enough data to evaluate the model's performance during tuning and with 20% allocated for testing, the evaluation provides a reliable indication of the model's ability to generalize. The training set is used to train the model, the testing set is used to evaluate the model's performance, and the validation set is used to adjust the hyper-parameters. To get the best performance out of the model and prevent overfitting or underfitting, hyper-parameter tuning is crucial. Also, Rainfall data often exhibits high temporal variability, making it prone to overfitting. The 60:20:20 split ensures that the model sees diverse scenarios during training and validation, reducing the likelihood of overfitting to specific trends or anomalies.

Implementation

Python programming language and the libraries such as Keras with TensorFlow, Pandas, NumPy, and Matplotlib were used to implement the prediction models (Bakker, 2017). The models underwent training on Google Colab using a free NVIDIA K80 graphics processing unit (GPU) with 12 GB of RAM from Google.

Performance evaluation

Common metrics for assessing the effectiveness of a machine learning model include Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and R^2. These metrics track the difference between the target variable's projected and actual values (Han et al., 2011). The average of the squared differences between the expected and actual values is called RMSE. Because it is easier to read in the same units as the target variable, it is frequently used to measure the average difference between the target variable's actual and expected values.

$RMSE = \sqrt{MSE}$ and $MSE = \left(\frac{1}{n}\right) \times \Sigma(y_i - \hat{y}_i)^2$

where y_i is observed points, $\hat{y}_i$ is predicted values and n is number of data points. MAE is the average of the absolute differences between the predicted and actual values. MAE measures the average absolute distance between the predicted and actual values of the target variable.

$MAE = \left(\frac{1}{n}\right) \times \Sigma|y_i - \hat{y}_i|$

where y is the actual value, ŷ is the predicted value, and n is the total number of data points. R^2 is a commonly used metric to evaluate the performance of predictive models and how well the model's predictions align with the actual observed values.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y}_i)^2}$$

where R^2 is the coefficient of determination, y_i represents the observed (actual) values of the dependent variable, $\hat{y}_i$ represent the predicted values of the dependent variable produced by the model, $\overline{y}_i$ is the mean (average) of the observed values y_i. The expected output of RMSE, MAE and R^2 depends on the specific problem being solved and the performance of the machine learning model. In the context of rainfall prediction models, these metrics help assess how well the model is performing in terms of accuracy and error magnitude

RESULTS

Analysis

Table 2 below shows the correlation between rainfall and other meteorological factors and indicates the strength and direction of the linear relationship between each meteorological attribute and rainfall. A positive correlation between a meteorological attribute and rainfall indicates that as the values of the meteorological attribute increase, there is a tendency for the rainfall to also increase. Table 3 below shows some samples of actual and predicted rainfall values using LSTM and Table 4 below shows some samples of actual and predicted rainfall values using GRU. Table 5 illustrates the prediction metrics values obtained with LSTM and GRU model. The RMSE, MAE and R^2 values obtained for both LSTM and GRU models are tabulated. All the metrics are evaluated for training data, validation data and test data. The GRU prediction model's metrics are obtained better in all the cases when compared to the metrics of LSTM prediction model.

Table 2. Correlation of meteorological factors with rainfall.

Max temperature	0.17
Min temperature	0.35
Relative humidity	0.10
Atmospheric pressure	0.05
Wind direction	0.08
Wind speed	0.05
Rainfall	1.00

Table 3. Samples of actual and predicted rainfall values using LSTM.

Actual Rainfall value in mm	Predicted rainfall value in mm
0.2	1.70
0.5	1.71
0	1.78
3.5	2.17
0.01	2.15
1.2	1.97
0	0
0	0
10.8	1.89

Table 4. Samples of actual and predicted rainfall values using GRU.

Actual Rainfall Value in mm	Predicted Rainfall Value in mm
6.42	4.99
2.07	1.78
2.40	3.97
0.01	2.08
0.7	1.70
0	1.92
0	0.01
0	0
1.71	0.5

Table 5. Evaluation metrics outcome.

Model	Training Error (RMSE)	Validation Error (RMSE)	Testing Error		
			RMSE	MAE	R^2
LSTM	10.15	9.92	2.51	1.79	0.81
GRU	10.08	9.90	2.31	1.51	0.95

Visualization

Python Matplotlib library was used to create a visualization of the results of a rainfall prediction model. By comparing the expected and real numbers, the resulting plot will enable us to visually evaluate the effectiveness of the rainfall prediction model.

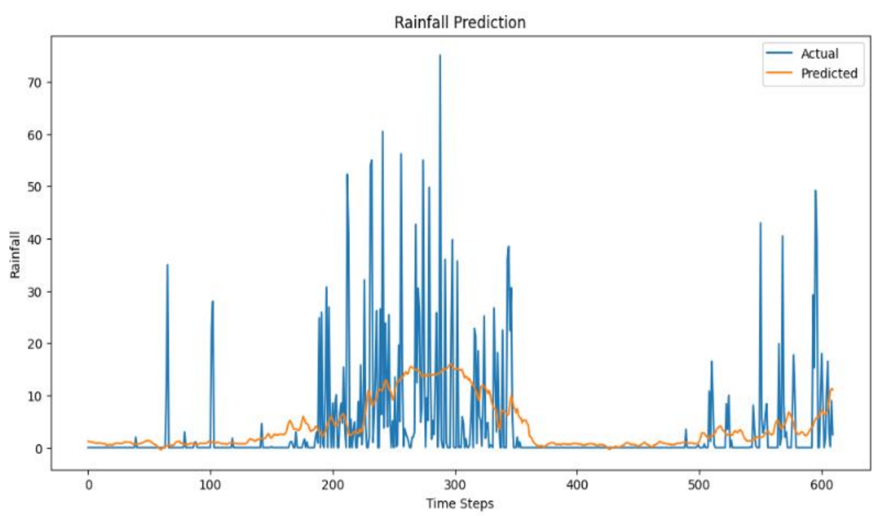


Fig. 2. Actual and predicted rainfall with LSTM.

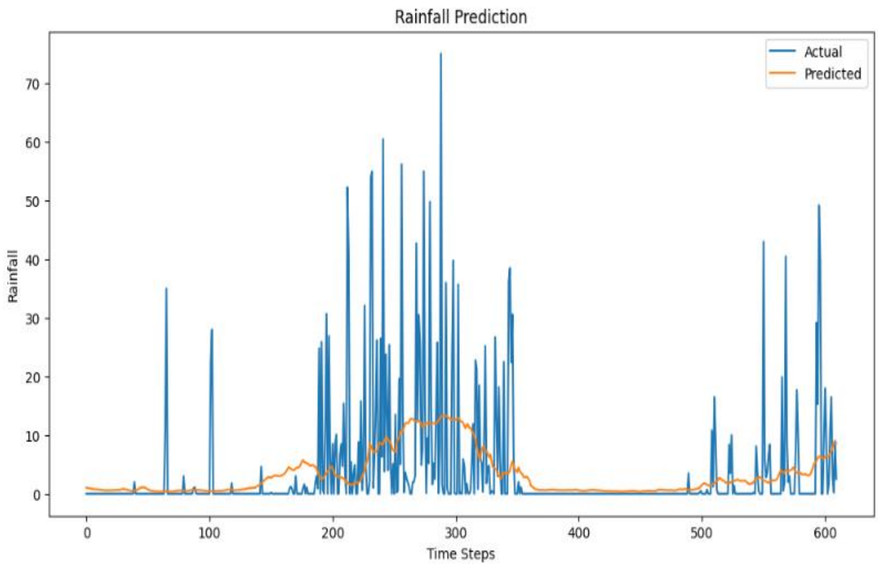


Fig. 3. Actual and predicted rainfall with GRU.

This plot is used to visually assess the performance of LSTM and GRU model by comparing its predictions with the actual values over time. It helps to identify patterns, trends, and discrepancies between the predicted and actual data. Here, x-axis represents the time steps i.e. days and y-axis represents the amount of rainfall. The plot visualizes how well the model's predictions (the 'Predicted' line) align with the actual rainfall values (the 'Actual' line) over the entire dataset, including training, validation, and test sets.

Performance analysis

The daily rainfall has been predicted using LSTM and GRU for 0.0001 learning rate, 50 epochs and 8 batch size. The training RMSE and validation RMSE of LSTM is found to be 10.05 and 9.92 respectively. Similarly, the training RMSE and validation RMSE of GRU is found to be 10.08 and 9.90 respectively. The difference of training and validation error of both the models are low, indicating that the models have learned the underlying patterns in the data without memorizing noise and can generalize well to new, unseen data. Test results for the GRU model show that, on average, they are more accurate than test results for the LSTM model in terms of both RMSE and MAE. The greater R^2 value of the GRU model suggests that a significant amount of the variability in the rainfall data is explained by the GRU model.

CONCLUSION

The purpose of this work is to perform daily rainfall prediction using meteorological dataset with different parameters and compare thus obtained results. Meteorological dataset with different factors obtained from Department of Hydrology and Meteorology, Babarmahal, Kathmandu, Nepal is used for the prediction and analysis task. For this prediction, LSTM and GRU models are used. While testing these two models, GRU outperformed LSTM with 6.16% better result with RMSE, 16.97% better result with MAE and 26.73% better with R^2. Rainfall prediction is an active area for research that may needs further efforts to improve accuracy due to several factors, including the complex and dynamic nature of the Earth's atmosphere. The accuracy may also vary when performed on different models, environment, parameters, and different epochs. By leveraging complex and hybrid models, prediction accuracy can significantly be improved in challenging scenarios such as high spatial variability, short-term extreme events, or long-term climate trends. Using complex hybrid models for rainfall prediction has the potential to provide significant benefits, particularly through improved accuracy, better handling of complex data, and adaptability to new trends.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Department of Hydrology and Meteorology, Babarmahal, Kathmandu, Nepal for providing data for this research. The authors would also like to thank all the faculty members and staffs of the Central Department of Computer Science and Information Technology and Central Department of Statistics, Tribhuvan University. Lastly, the author wishes to express their sincere appreciation to the anonymous reviewers for their meticulous review of the paper.

CONFLICT OF INTEREST

The authors confirm no conflict of interests.

AUTHOR CONTRIBUTION

MP led the conceptualization of the research, designed the methodology, collected data and prepared the first draft of the manuscript. NP reviewed the literature, performed result analysis, interpreted results, supervised the research, provided critical revisions to the manuscript, edited and prepared the final manuscript.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

DATA AVAILABILITY

The data supporting the findings of this study are available upon reasonable request from Department of Hydrology and Meteorology, Babarmahal, Kathmandu, Nepal. The data are not publicly available.

ETHICAL STATEMENT

This research was conducted in accordance with established ethical guidelines and principles. The authors declare that there are no conflicts of interest influencing the outcomes of this research. This research adheres to principles of honesty, transparency, and academic integrity.

REFERENCES

Aswin, S., Geetha, P., & Vinayakumar, R. (2018). Deep learning models for the prediction of rainfall. *Proceedings of the 2018 IEEE International Conference on Communication and Signal Processing, ICCSP 2018*. https://doi.org/10.1109/ICCSP.2018.8523829

Bakker, I. D. (2017). *Python deep learning cookbook*. Packt Publishing.

Chhetri, M., Kumar, S., Roy, P. P., & Kim, B. -G. (2020). Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan. *Remote Sensing*, *12*(19), 3174. https://doi.org/10.3390/rs12193174

Cho, K., Merrienboer, B. v., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computer Linguistics. https://doi.org/10.3115/v1/D14-1179

Endalie, D., Haile, G., & Taye, W. (2022). Deep learning model for daily rainfall prediction: case study of Jimma, Ethiopia. *Water Supply*, 22(3). https://doi.org/10.2166/WS.2021.391

Géron, A. (2017). Hands-on machine learning with scikit-Learn and tensorflow. O'Reilly Media, Inc.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann Publishers Inc.

Hassan, M. M., Rony, M. A. T., Khan, M. A. R., Hassan, M. M., Yasmin, F., Nag, A., Zarin, T. H., Bairagi, A. K., Alshathri, S., & El-Shafai, W. (2023). Machine learning-based rainfall prediction: Unveiling insights and forecasting for improved preparedness. *IEEE Access*, *11*, 132196-132222. https://doi.org/10.1109/ACCESS.2023.3333876

Hess, P., & Boers, N. (2022). Deep learning for improving numerical weather prediction of heavy rainfall. *Journal of Advances in Modeling Earth Systems*, *14*(3). https://doi.org/10.1029/2021MS002765

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Lenderink, G., & Fowler, H. J. (2017). Hydroclimate: Understanding rainfall extremes. *Nature Climate Change*, *7*, 391-393. https://doi.org/10.1038/nclimate3305

Muhammad, A. U., Li, X., & Feng, J. (2019). Using LSTM GRU and hybrid models for streamflow forecasting. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, *294 LNCIST*. https://doi.org/10.1007/978-3-030-32388-2_44

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A Comparison of ARIMA and LSTM in forecasting time series. *Proceedings. 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*. https://doi.org/10.1109/ICMLA.2018.00227

Surta, W., Basuki, K. T., Negara, E. S., & Kunang, Y. N. (2023). Rainfall prediction in Palembang city using the GRU and LSTM methods. *Journal of Data Science*, *4*(2), 1-13. http://eprints.intimal.edu.my/1730/

Zargar, S. A. (2021). *Introduction to Sequence Learning Models: RNN, LSTM, GRU*. https://doi.org/10.13140/RG.2.2.36370.99522

Reference to this paper should be made as follows:

Pujara, M., & Paudel, N. (2024). Rainfall prediction using long short-term memory and gated recurrent unit with various meteorological parameters. *Nep. J. Stat, 8,* 47-60.