A NOVEL MULTI-HEADED TRANSFORMER-DRIVEN RAINFALL FORECASTING BASED ON SEQUENTIAL WEATHER PARAMETERS

[1]Hariharan S and [2]Dr. Prakasam S

[1]Research Scholar, Department of Computer Science and Applications,
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram, Tamilnadu,
631561 – India.

[2]Associate Professor, Department of Computer Science and Applications,
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram, Tamilnadu,
631561 – India.

Abstract- Rainfall forecasting continues to be a daunting task for the Meteorological Departments and many private Weather prediction agencies around the world. The vast expanse of the parameters that control the real-time weather patterns is making the weather models more vulnerable to failure and error-prone. The Weather parameters are highly correlated both spatially and temporarily. To capture interdependencies of the weather parameters and to contextualize the favorable conditions for the occurrence of rainfall, we put forth a novel method of using Multi-headed Transformer for analyzing contextual correlations of weather parameters against the rainfall occurrence. This paper treads on a novel approach of using the state-of-the-art Transformer technology to unearth the latent inter-relations between weather parameters and rainfall. A novel Weather to Vector representation is being suggested for representing the input weather parameters as weights of the neural network which acts as the input embedding for the Transformers.

Keywords: Weather to Vector Representation, Transformers, Rainfall forecasting, Time series sequence classification.

1. Introduction

Accurate and timely weather forecasting still remains elusive and challenging around the world. As adverse weather conditions wreak havoc globally, prediction of rainfall becomes paramount for taking precautionary measures so that human lives can be saved and economic losses can be avoided. Analyzing the causal relationship between various parameters and its effect on the occurrence of rainfall is pivotal to gaining valuable insights on weather patterns. In order to realize the set objectives, the latest cutting-edge technology that is prevalent in the field of Time series analysis viz. Transformers are used to create a robust weather model. This novel approach tries to leverage the power of Transformers in better prediction of rainfall. It harnesses the efficiency of multi-headed self attention mechanism to focus on a specific time window and to isolate and attend to the causal relationship between individual parameters and the rainfall occurrence. As the Transformers takes vector values as input, a novel procedure called Weather2Vec representation is used for embedding the weather parameters as vector representations. The Skipgram algorithm is tweaked and used to train the weights of neural network which actually acts as the vector embedding that are fed as input to Transformers.

A. Problem Statement:

Given the time series weather data comprising of Temperature, Humidity, Precipitation, Cloud cover and Wind speed, the objective is to propose a Transformer-driven Weather model for better forecasting of the occurrence of rainfall for any given day and enhancing the robustness of the model by unearthing the latent features of the weather parameters and attending to the most favorable weather conditions that causes the rainfall.

B. Scope:

The research has profound scope in the area of Disaster Management and Agriculture sector as it helps them to be better informed about the probability of occurrence of the rainfall activities and also better prepared. It enhances the robustness of the rainfall prediction and limits the economic losses of the farmers.

C. Multi-headed Self-attention Transformers:

Transformers are predominantly used in NLP models. In this paper a new path is explored for analysing how Transformers can be better used to forecast Weather. Transformer architecture has the following features:

1. **Self-Attention Mechanism**: This is the key component of transformers. It allows the model to weigh the importance of each word in the input sequence against every other word, capturing dependencies and relationships without regard to their position in the sequence.
2. **Encoder-Decoder Structure**: Many transformer-based architectures, like the original Transformer model, consist of an encoder and a decoder. The encoder processes the input sequence, while the decoder generates the output sequence.
3. **Multi-Head Attention**: To enhance the learning capacity and effectiveness of attention mechanisms, transformers often employ multi-head attention. This means that attention is computed multiple times in parallel, each with different sets of learned weights.
4. **Feedforward Neural Networks**: After the attention mechanism, transformers typically include feedforward neural networks (FFNs) at each position in the network. These FFNs are usually fully connected layers that process the information extracted by the attention mechanism.
5. **Positional Encoding**: Since transformers do not inherently understand the order or position of the input tokens, positional encoding is added to provide information about the position of each token in the sequence.

6. **Normalization Layers**: Layer normalization or batch normalization is used to stabilize and accelerate the training of deep neural networks.
7. **Residual Connections**: Residual connections (skip connections) are often used to help with the flow of gradients during training and to facilitate learning deeper architectures.
8. **Output Layer**: Depending on the task, the output layer of a transformer-based model could vary. For example, in language translation tasks, it might be a linear layer followed by a softmax activation for predicting the next word in the target language.

These components together enable transformers to achieve state-of-the-art performance on various natural language processing tasks like machine translation, text generation, sentiment analysis, and more. They represent a significant advancement in deep learning architectures, particularly in handling sequential data where capturing long-range dependencies is crucial.

2. Literature Survey

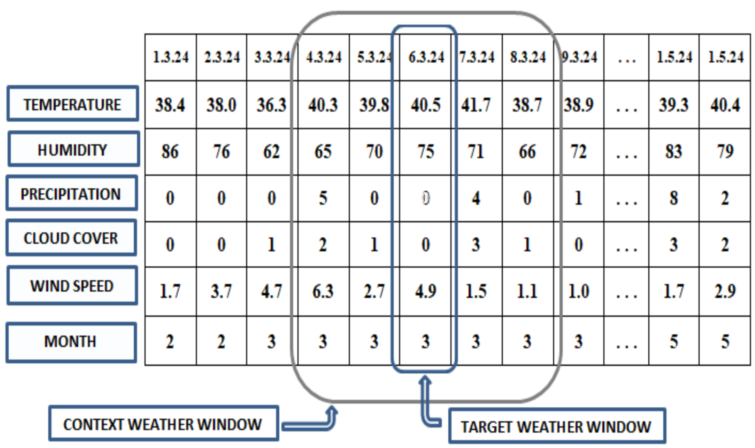
Research on weather prediction using transformer models is an evolving area, leveraging the strengths of transformers in handling sequential data and capturing long-range dependencies. Here are some relevant research articles and papers:

Various machine learning approaches are being used, especially transformer-based models, for precipitation nowcasting, which has proved to be quite beneficial and has profound scope in weather prediction [1]. Studies that focus on solar radiation, is employed in transformer models for time series forecasting, demonstrating widespread applicability to weather-related variables [2]. Various deep learning approaches exists, including transformers, for estimating precipitation from radar data, which has been very crucial for weather prediction.[3]. Deep Transformer models have been very effective in time series forecasting such as Influenza prevalence which can also be applicable to weather variables [4]. Multiple Transformer-based models have been successfully implemented for undertaking time series forecasting, showing potential applications in weather forecasting which has shown tremendous potential for accurate prediction of the weather patterns [5]. Studies have analysed a benchmark dataset named WeatherBench, and evaluation platform by implementing weather forecasting models, using transformer-based approaches [6]. Various studies has been explored for analysing the application of deep transformer models for time series forecasting, specifically focusing on weather prediction tasks [7]. Studies have Focused on improving weather prediction accuracy using deep transformer models, highlighting their capability to handle large-scale weather datasets [8]. Transformer-Based Time Series Forecasting for Renewable Energy and Weather Data have been extensively studied which Discusses the application of transformer models for time series forecasting, including weather data relevant to renewable energy forecasting [9]. Transformer-Based approaches have used for analyzing Spatio-Temporal patterns in order to forecast of Rainfall [10].

3. Proposed Rainfall Prediction Model Using Transformers:

Weather2Vec representation:

Weather2Vec representation is a novel method to embed the weather parameters using Skipgram model. The model is trained by feeding the time series data consisting of weather parameters viz. Temperature, Humidity, Precipitation, Cloud cover and Wind Speed as input. The Target window contains the weather parameter values for any given day and the context window refers to 3 days before and 3 days after the target window. The output is recorded as 4 categorical values viz. Heavy Rain, Moderate Rain, Drizzle and No Rain. For the given target window containing the weather parameters, the occurrence of rain is checked for the days spanning the context window. Accordingly the output is set to one of the four categories. The neural network weights are trained using backpropagation for the given target weather parameters with respect to the occurrence of rain in the context window. The optimized weights and biases minimize the error on the task of neural network to predict the occurrence of rainfall given the target weather parameters. The output is a probability distribution vector which predicts the probability for occurrence of rainfall in the context window. The predicted probability will be high for rainy days which share the same context and low for non-rainy days which again share the same context.



	1.3.24	2.3.24	3.3.24	4.3.24	5.3.24	6.3.24	7.3.24	8.3.24	9.3.24	…	1.5.24	1.5.24
TEMPERATURE	38.4	38.0	36.3	40.3	39.8	40.5	41.7	38.7	38.9	…	39.3	40.4
HUMIDITY	86	76	62	65	70	75	71	66	72	…	83	79
PRECIPITATION	0	0	0	5	0	0	4	0	1	…	8	2
CLOUD COVER	0	0	1	2	1	0	3	1	0	…	3	2
WIND SPEED	1.7	3.7	4.7	6.3	2.7	4.9	1.5	1.1	1.0	…	1.7	2.9
MONTH	2	2	3	3	3	3	3	3	3	…	5	5

Fig.-1 Context window and Target window in Weather2vec model.

Probability to get rain in the context days given target weather parameters is represented by $P(W_{t+j} \mid W_t)$

The Loss function used for training the neural netework is given by

$$J = -\sum_{t=1}^{T} \sum_{-m \leq j \leq m} \log(P(W_{t+j} \mid W_t))$$

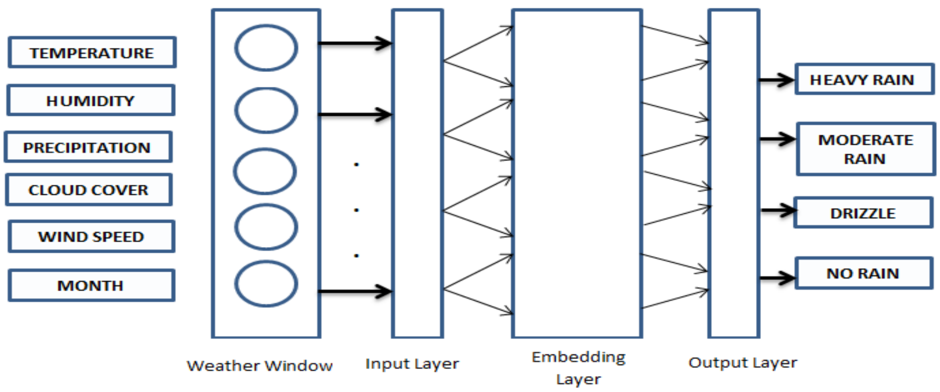


Fig.-2: Skipgram model for Weather2vec representation.

The Softmax activation function provides probability distribution as given below:

P(Rain in Context days | Target weather parameters)

$$= \exp(U^T_{target} V_{context}) / \sum_{w=1}^{days} \exp(U^T_{target} V_w)$$

As shown in the figure-2, skipgram model is used for converting the input weather parameters into the vector embedding.

After training of the neural network, the trained weights are taken as the final embedding values of Weather2Vec model.

Encoding Positions to the Weather Embedding:

The time series data of Weather parameters for N-days are processed into word embedding $Z^w : N \times h^w$, where N represents the number of days in the context window and h^w represents the embedding size.

Positional Embedding is given by

$$PE_{k,2i} = \sin(k / (10000^{2i/hw}))$$

$$PE_{k,2i+1} = \cos(k / (10000^{2i/hw}))$$

Where k is the position of the day spanning the sequence [0, N-1]

i is the index spanning the dimension of Weather embedding in $[0, h^w - 1]$

Once the Positional encoding is calculated, it is added with the Weather embedding as shown below:

$$Z^{w}_{New} = Z^{w} + PE$$

As shown in figure-3, the positional encodings are calculated for the time series data using the above formula that incorporates sinusoidal waves, which is then added with vector embedding to produce the final embedding values.

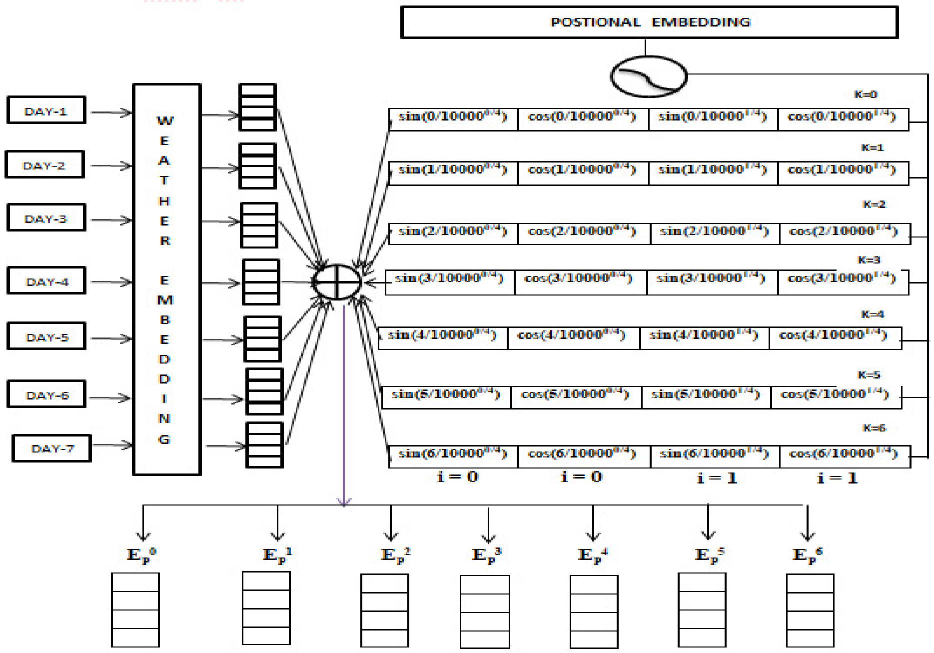


Fig.3 – Positional Encoding of Weather parameters

Multi Head Self-Attention Mechanism in Rainfall Prediction:

The attention mechanism is applied on the weather parameters to determine the major parameters affecting the occurrence of rainfall for a given time period. The self-attention mechanism tries to unearth the most favorable parameters that caused the rainfall by attending to the historical data for a given season. Multi Head Self-attention refers to analyzing each of the weather parameters individually for its impact on the positive occurrence of the rainfall by focusing on a specific weather window in the yesteryears. There are multiple heads for each of the weather parameters such as Temperature head, Humidity head, Precipitation head, Wind head and Cloud cover head. Each head ascertains how much of an effort does each of the respective weather parameter have on the rainfall activity for the given window of time period.

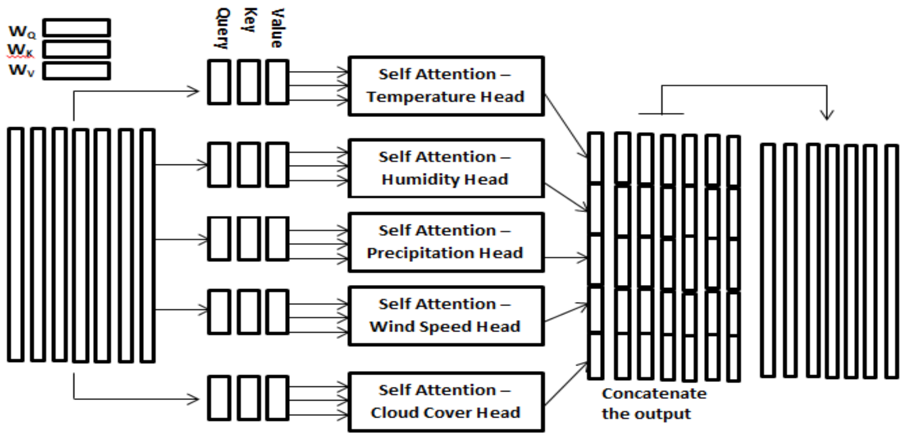


Fig.-4: Multi-headed Self attention mechanism

In self-attention mechanism, each input vector-x_i is subjected to three linear transformations governed by three weight matrices namely, the Query Wq, the Key W_k and the Value W_v as shown in the figure-4.

The Query matrix W_q represents the learned weights of a neural network which is trained for the occurrence of rainfall for any specific month. The neural network is trained for the the respective months separately for obtaining individual weight matrices W_q.

The Key matrix W_k represents the learned weights of neural network which is trained for the occurrence of rainfall for a specific weather parameter. Each Self-attention head corresponding to the individual weather parameters is associated with a separate Key matrix referring to the learned weights considering only the specific weather parameters for which it is being trained upon.

The self-attention mechanism is implemented as follows:

$$Q_i = W_q X_i \qquad K_i = W_k X_i \qquad V_i = W_v X_i$$

The W_q matrix performs a linear transformation on the input vector X_i to produce a vector Q_i, which actually attends to the rainfall trends which prevailed in the current months of yesteryears for the given weather parameters.

The W_k matrix performs a linear transformation on the input vector X_i to produce a vector K_i, which actually attends to the rainfall trends based on the specific weather parameter corresponding to its Attention head.

$$W'_{ij} = Q_i^T K_j$$

Taking the dot product of query vector Q_i and Key vector K_i, with which their similarity index can be measured. Q_i contextualize the rainfall inducing weather parameters for the given month in the yesteryears and K_i contextualize the rainfall induced by a specific weather parameter alone. The similarity measure helps to determine if the prevalent weather parameters are contextually similar to the rainfall-inducing parameters of the past. The attention weights are normalized using the softmax function.

$$W_{ij} = \text{Softmax}(W'_{ij})$$

Finally the attention score is calculated as follows:

$$Y_i = \sum_j W_{ij} V_j$$

As shown in Figure- 5, the Query, Key and Value matrices are calculated for the input X_i, each attention head produces a different output vector Y_i which are concatenated and passed through a linear transformation to reduce the dimension back to K.

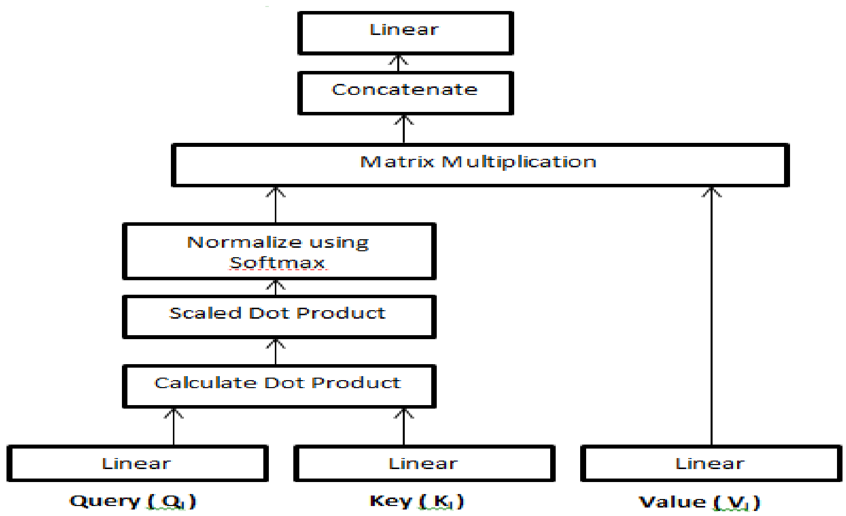


Fig.-5 Computation of Query, Key and Value Matrices

4. Rainfall Prediction Algorithm using Transformers:

Step-1: The Input Weather parameter values are converted to Vector embedding using Weather2Vector embedding scheme.

Step-2: Positional embedding is generated for each input embedding based on the sequence of the days as given in the equation below.

$$\mathbf{PE_{k,2i} = sin(k / (10000^{2i/hw})}$$

$$\mathbf{PE_{k,2i+1} = cos(k / (10000^{2i/hw})}$$

The input embedding and positional embedding is added together to produce the final embedding.

$$\mathbf{Z^w_{New} = Z^w + PE}$$

Step-3: The embedding vectors are fed to the respective Attention-head block corresponding to the individual weather parameters. The embedding vectors are then subjected to the self-attention mechanism based on Query, Key and Value weight matrices in accordance with

Attention-head. The attention scores are calculated and normalized to produce the output vector.

$$\mathbf{head_i = Attention(QW_i^q, KW_i^k, VW_i^v)}$$

Step-4: The outputs of multi-head transformers are averaged to produce a single vector. The vector is softmaxed to produce probabilities for the occurrence of rainfall for the given input Weather parameters.

$$\mathbf{Multihead(Query, Key, Value) = Concat\ (head_1, head_2, head_h)\ W_i^o}$$

Step-5: The probability value is compared with the target value to estimate the loss and accordingly the weight values of query, key and value matrices are adjusted and the model is trained.

Step-6: After training the weight matrices of each attention-head, the weather parameters are provided as input and the sequence classifier produces the probability of rainfall occurrence for any day.

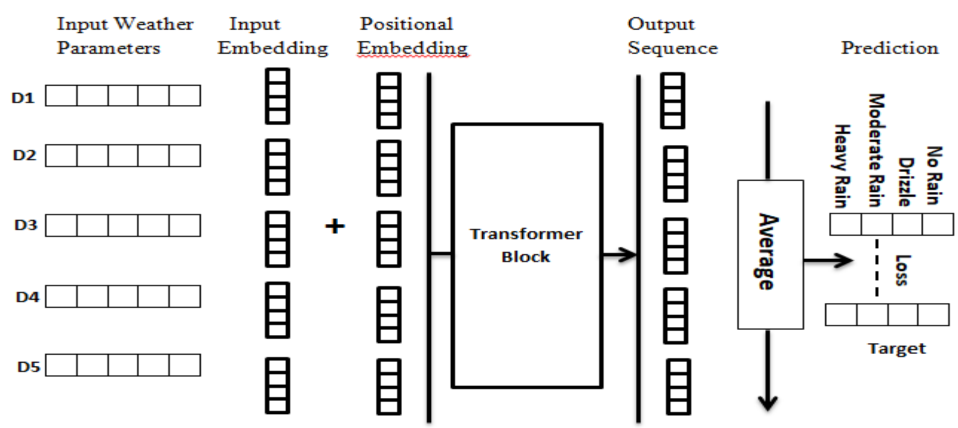


Fig.-6: Architecture for Rainfall Prediction

5. Experimental Analysis:

We compared a Transformer model along with two benchmark models (LSTM and ANN) to perform simulated predictions of rainfall occurrence as depicted in figure-7. The Transformer model achieves the best performance, with an NSE greater than 0.9 and RMSE and MAE values. The ANN model exhibits inferior performance compared to the other two models, with calibration and validation NSE values.

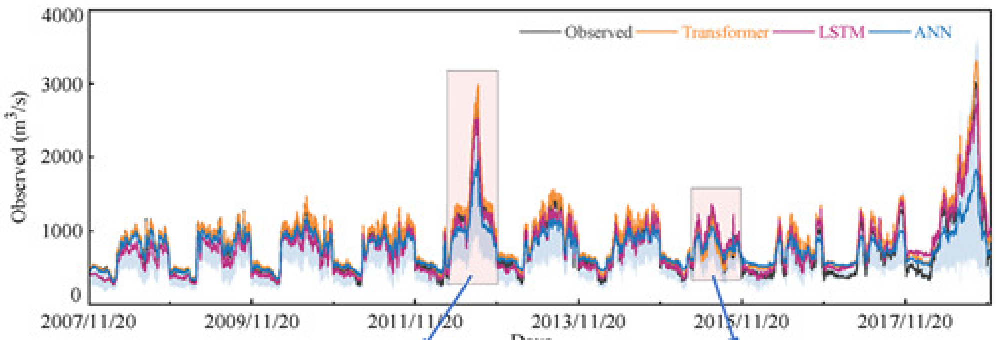


Fig.-7: Simulated Prediction of Rainfall occurrence

We present an analysis of the scatter density distributions of the simulated and observed values for all models. The results obtained during the validation are shown in Figure-7. For all models, the simulated values exhibit a significant correlation with the observed values. The model has shown improved performance of Transformer compared to LSTM, and ANN models

The scatter plot of the BS-Former model shows that the simulated and observed values at all sites are closely distributed near the 45-degree line with minimal deviation, demonstrating its optimal performance. In comparison, Transformer model exhibits an approximate improvement of 0.03 in the NSE index, a decrease of approximately 0.02 in the RMSE, and a reduction of approximately 0.01 in the MAE. The scatter plot distribution of the LSTM model lies generally below the 45-degree line, but the introduction of the base flow separation technique moves the results of the BS-LSTM model closer to the 45-degree line, with an average increase of approximately 0.05 in the NSE index. Conversely, the ANN model demonstrates a significant bias, as its scatter plot distribution indicates overestimation of low values and underestimation of high values.

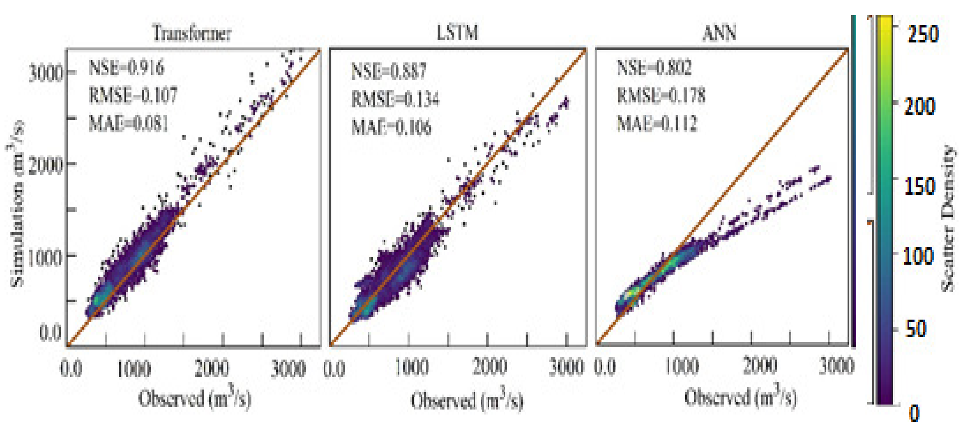


Fig.-8: Scatter plot distribution

To explore the potential improvement in the forecasting ability of different models, we analyzed the simulation performance of all models for lead times of 1–7 days, focusing using all the models during the validation period. The performance indicators are shown in Figure-9. The results suggest that there is noticeable consistency in the performance of the different models. As the lead time increases, the simulation performance of all models tends to exhibit a generally decreasing trend. The Transformer model demonstrates superior forecasting capabilities at longer lead times. Specifically, the performance of the Transformer model for runoff prediction with a 7-day lead time is equivalent to that of the LSTM and ANN models for 4-day and 3-day lead times, respectively.

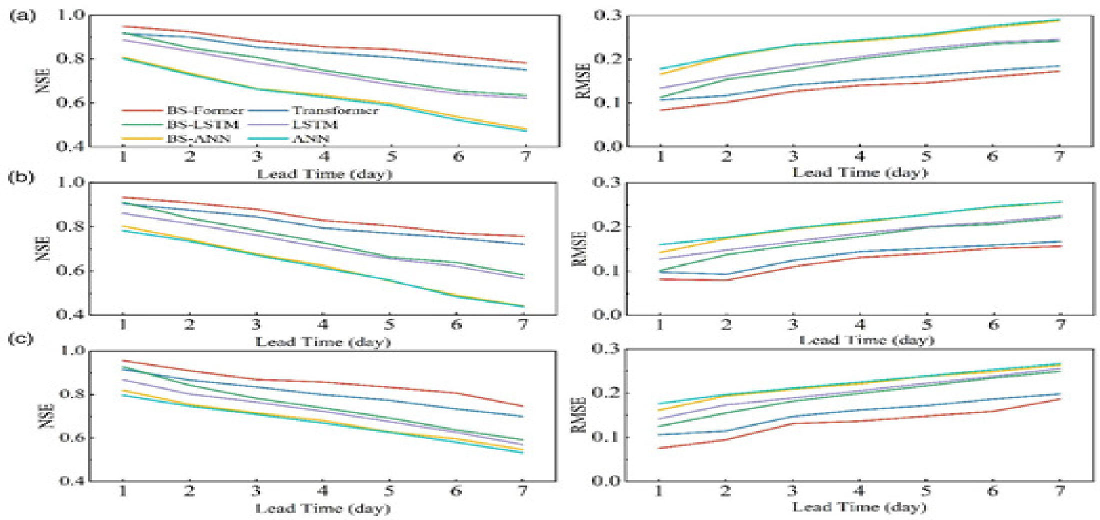


Fig.- 10: Simulated Flow Distribution for all models.

The simulated flow distribution for all models with different lead times is shown in Figur-10. The results show that the overall distribution of the results of the ANN model is too concentrated and unstable at different lead times compared to the actual observed values,. The LSTM model underestimates the flow value when the lead time is greater than 2 days. In comparison, the fluctuations of the Transformer model are relatively stable, and the distribution characteristics of the simulated values are consistent with the observed data at different lead times. On some occasions, the Transformer model overestimates the flow values a little bit.

6. Conclusion:

This study has implemented a novel Weather2Vec embedding for representing the input weather parameters which is fed as input to the Transformer. A Transformer based algorithm has been proposed and implemented for prediction of rainfall occurrence given a time series data containing weather parameters. The performance of the proposed Transformer-based algorithm has been compared with LSTM and ANN models, which has proven that Transformer model offers greater flexibility in predicting long-term correlations in sequential data owing to its self-attention mechanism. Moreover, it exhibits significant generalizability and stability for weather modelling, with both calibration and validation NSE values exceeding 0.9.

REFERENCES

[1] Stephan Rasp, Peter Dueben, and Nils Thuerey (2020). “Machine Learning for Precipitation Nowcasting: A Benchmark and Comparative Study", Geoscientific Model Development Discussions.

[2] Christoforos Anagnostopoulos, Alexandros Doulamis, Nikolaos Doulamis, and Anastasios Koutras, (2021) "Machine Learning for Time Series Forecasting of Global Radiation", Remote Sensing

[3] Sebastian Ertel, Tobias Haist, Jonas Hirsch, and Stefan Zachow,(2021) “Deep Learning for Precipitation Estimation from Radar Images", Remote Sensing

[4] Pavlo Mozharovskyi, Patrick Schwab, and Bernhard Schölkopf. (2020), "Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case", arXiv preprint arXiv:2001.08317, 2020.

[5] Boris Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio (2020) "A Transformer-based Approach for Time Series Forecasting", Advances in Neural Information Processing Systems (NeurIPS).

[6] Stephan Rasp, Peter D. Dueben, Soukayna Mouatadid, et al. (2021), "WeatherBench: A Benchmarking Platform for Data-Driven Weather Forecasting", Geoscientific Model Development, Volume 14, Issue 9, September 2021.

[7] Tianyu He, Gregor Urban, Ian Goodfellow (2021). "Deep Transformer Models for Time Series Forecasting: The Case of Weather Prediction". NeurIPS Workshop on Tackling Climate Change with Machine Learning, December 2021.

[8] Sujitha H, Saranya S, (2021). “Predicting Weather Data Using Transformers”, International Conference on Computer Communication and Informatics (ICCCI).

[9] Marios Protopapas, Tommi Jaakkola (2021). “”Transformer-Based Time Series Forecasting for Renewable Energy and Weather Data”, IEEE International Conference on Artificial Intelligence and Statistics (AISTATS).

[10] Hongzhi Shi, Yaqian He, et al.(2021), "Transformer-Based Spatio-Temporal Forecasting of Rainfall", Journal of Hydroinformatics

[11] Yanhua Li, Xuechen Li, et al.(2022), "Rainfall Forecasting Using Multi-Horizon Transformer Model" Springer International Journal of Machine Learning and Cybernetics.