Time Series Visualization using Transformer for Prediction of Natural Catastrophe

Shivam Pandey[1], Mahek Jain[2]

[1]Department of Computer Science, University of California, Davis, CA – 95616, USA
Email id: *sspandey[at]ucdavis.edu*

[2]Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering (RVCE), Bangalore – 560059, India
Email id: *mahekjain.cs18[at]rvce.edu.in*

Abstract: ***The extension of the forecast time is an essential requirement for real-world applications, which includes early caution for severe climate conditions. In this paper, we come up with a new approach to time series forecasting. The time-series data is generic in lots of disciplines and engineering. Time series prediction is a vital assignment in time-series data modeling and is an important area of deep learning. We have developed a novel technique that makes use of a Transformer-based deep learning model for the prediction of time-series data. This technique works with the aid of self-attention mechanisms to study complicated patterns and dynamics from time-series data. Moreover, it is a preferred framework and may be implemented in univariate and multivariate time series data, in addition to time series embedding. Using natural disasters such as flood forecasting as a case study, we show that the forecast outcomes produced using our technique are similar to the state-of-the-art.***

Keywords: Weather Forecasting, Transformer Networks, Time Series, Deep Learning, Attention Mechanisms

1. Introduction

Flood forecasting enables flood managers to predict with high precision when local floods are likely to occur. The overall performance of flood forecasting models should be evaluated periodically and a continuous cycle recalibration of the model should be performed after a new flood event to offer an up-to-date, beneficial, and accurate flood forecasting tool. River flooding can cause loss of life, devastating property damage, and adverse economic and environmental impacts. Although flood risks cannot be completely ruled out, real-time flood forecasting models, as an essential part of a flood caution service, can assist in providing flood warnings to the public in a timely manner with enough time to minimize the damage caused by flooding. This research uses time series prediction [18] with Transformer, a deep learning model, to predict floods in Australia. The aim of time series forecasting is to predict the potential values of a time series based on its historic values.

It is now generally accepted that flood risks may be decreased however by no means absolutely eliminated. Therefore, in latest years there was a big improvement from a flood protection method to a flood threat control strategy; Therefore, it's far vital to offer a powerful and dependable flood caution service as a part of the flood threat control process. Flood warnings must be furnished with an affordable wait time to the general public and emergency offerings so that you can take steps to reduce flood damage.

In this work, we've advanced a novel approach to time series forecasting primarily based on the Transformer architecture. On contrary to sequence-oriented models, Transformer no longer processes data in an orderly manner. Rather, it processes the whole data stream and makes use of self-attention mechanisms [17] to analyze the dependencies within the data stream. Thus, transformer-based models have the capacity to model complex dynamics of time-series data which can be challenging for sequence models. In this paper, we display that a transformer-based model can be efficiently carried out to the time series prediction function and outperforms many existing prediction techniques.

2. Literature Survey

The survey that has been conducted consists of the existing technology and studies associated with the topic of our project. It is an attempt to apprehend the efforts which have been made into this area of study, and additionally to apprehend where our efforts ought to be centered at the time of developing this project.

Due to the great importance of time series prediction, a variety of models have been developed. Most of the time series forecasting methods make use of some of the classical tools [15] like ARIMA, RNN, etc. ARIMA [1] solves the prediction problem by converting non-stationary processes into stationary processes through differentiation. A filtering method [7] has also been introduced for series prediction [4]. In addition, the recurrent neural network (RNN) model was used to model time dependencies for time series [10]. DeepAR [12] combines autoregressive and RNN methods to model the probabilistic distribution of future series. LSTNet [8] puts together the convolutional neural networks (CNNs) along with recurrent skip connections in order to capture both short and long-term temporal patterns. RNNs involving the attention layer [14] introduce temporal attention for exploring the long-term dependencies for forecasting. Moreover, many projects based on temporal convolution network (TCN) [13] try to model temporal causality with causal convolution. This in-depth predictive model focuses primarily on modeling temporal relationships using repeated connections, temporal attention, or causal convolution. Recently, transformers based on self-attention mechanisms have shown great power in sequential data, such as natural language processing [2], audio processing [3], and also computer vision [5].

Shengdong Du [6] developed an encoder-decoder model with temporal attention in order to solve the problem of multivariate time series forecasting. This model is based on Bi-LSTM layers to learn long-term dependency. [9] Bryan Lim adopted a new attention-based architecture named Temporal Fusion Transformer (TFT). It makes use of recurrent layers and self-attention layers enabling high performance. R Mohammdi Farsani [11] proposed a method to make the predictions at larger intervals more accurate. The transformer network model can provide desired results with a better network configuration and better adjustment of attention.

3. Objectives

We have advanced a general Transformer-based model for predicting time series. The paper shows that our method is better than the other models like Recurrent Neural Network (RNN). It has been shown that the Transformer-based model can accurately predict natural calamities like floods using various functions. The work of the Transformer-based model proves to achieve state-of-the-art forecasting results.

4. Transformer Model

4.1. Methodology

The primary concept supporting the Transformer model is self-attention: it is the capacity to pay attention to different positions in an input sequence to compute an illustration of that sequence. Transformers create stacks of personal attention levels. The self-attention block encodes each new input concerning all other previous inputs, placing the accent based on the relevance calculation with respect to the current token.

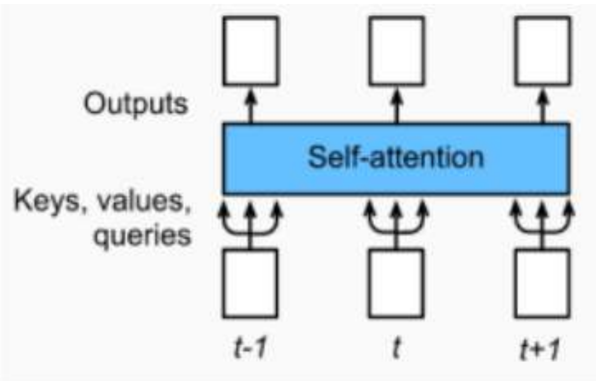

Figure 1: Illustration of self-attention architecture encoding the new inputs

The input to the transformer is time series specified. For each new input, the model returns a new estimate for the next timestamp. The transformer model is described as

1) Multi-Head Attention: The multi-head attention level consists of h parallel attention levels. Each layer is called a head. For each head, we use three solid layers with hidden dimensions p_q, p_k, and p_v to project their respective queries, keys, and values, before feeding the attention layers. The outputs of these h heads are combined and then projected from another solid layer.
2) Position-wise Feed Forward: A position-wise feed-forward network accepts three-dimensional input with shapes (batch size, sequence length, characteristic size). It consists of two solid layers that apply to the last dimension, which means that the same solid layer is used for each positioning element in the sequence, therefore it is called position-wise. Along the lines of multi-head attention, a positional feed-forward network will only change the final dimension of the input. Also, if the two elements in the input sequence are identical, the corresponding outputs will also be identical.
3) Positional Encoding: Both the multi-head attention level and the positional feed-forward network calculate the output of each element in the sequence independently. This property allows us to compute in parallel but is inefficient for modeling sequence information. The transformer model then adds position information in the input sequence.

4.2. Model Architecture

The proposed Transformer-based prediction model is developed along the lines of the original Transformer architecture inclusive of encoder and decoder layers.

Paper ID: SR211022155010 DOI: 10.21275/SR211022155010

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

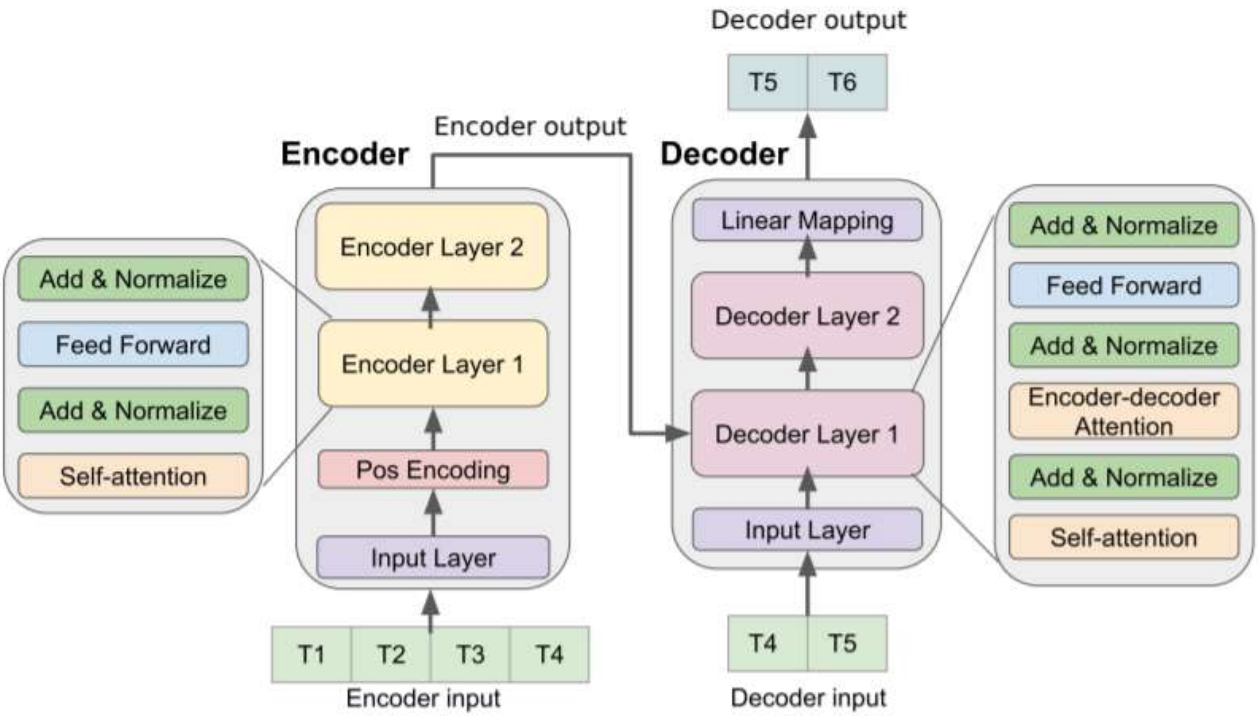

Figure 2: Transformer-based model architecture depicting the encoder and decoder layer

Encoder: The encoder includes an input level, a positional encoding level, and a stack of four similar encoder levels. The input layer maps the input time-series data to a dimensional vector d_{model} across a fully connected network. This step is important for the model to use the multi-headed attention mechanism. Position coding with sine and cosine functions is used to encode sequential information in time-series data by adding an input vector element to the position-coding vector. The output vector is then fed into a four-level encoder. Each encoder level consists of two sublevels: a self-attention sublevel and a fully connected feedforward sublevel. Each sublevel is accompanied by a normalized level. The encoder generates a d_{model} dimensional vector to be fed to the decoder.

Decoder: We use a decoder design identical to the original transformer architecture. The decoder additionally includes an input layer, four identical decoder layers, and an output layer. The decoder input starts with the last data point of the input passed to the encoder. The input layer maps the decoder input to a d_{model} dimensional vector (Wu, N., et. al, 2020). Along with the two substrates in each encoder layer, the decoder inserts the third substrate in order to apply a self-attention mechanism to the encoder output. Finally, there is an output layer that maps the output of the last decoder level to the target timeline. We use lookahead masking and offset the position between the decoder input and the destination output within the decoder to make sure that the data point prediction in the time series depends only on the previous data point.

5. Research Work

After training the model, we try to predict one day into the future. The graph below shows the training target, validation target, and test target.

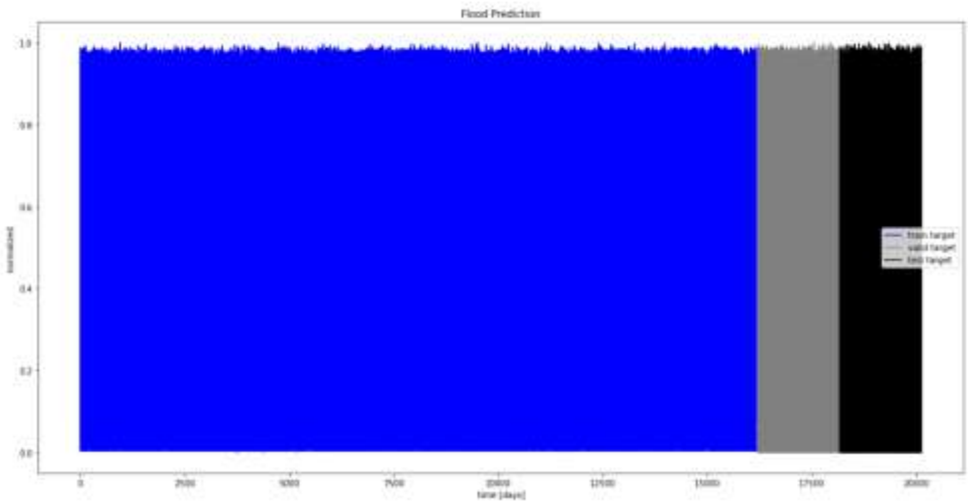

Figure 3: Plot showing the train, valid, and test target

The original model was trained on 200 epochs, and high accuracy was achieved with training_loss: 0.0015 and val_loss: 0.0012.

6. Experimental Results

6.1 Experimental Data Set

The experimental data set that has been used in the paper is the flood.csv which was extracted from Kaggle and consists of approximately 150,000 entries from various cities in Australia. Flood forecasts are made based on many factors such as minimum/maximum temperature, Evaporation, Sunshine, Wind Gust direction and speed, Humidity, Pressure, Clouds, etc. It consists of the data for the years 2008 to 2020.

6.2 Performance Analysis

The time series visualization for the original transformer model is shown below:

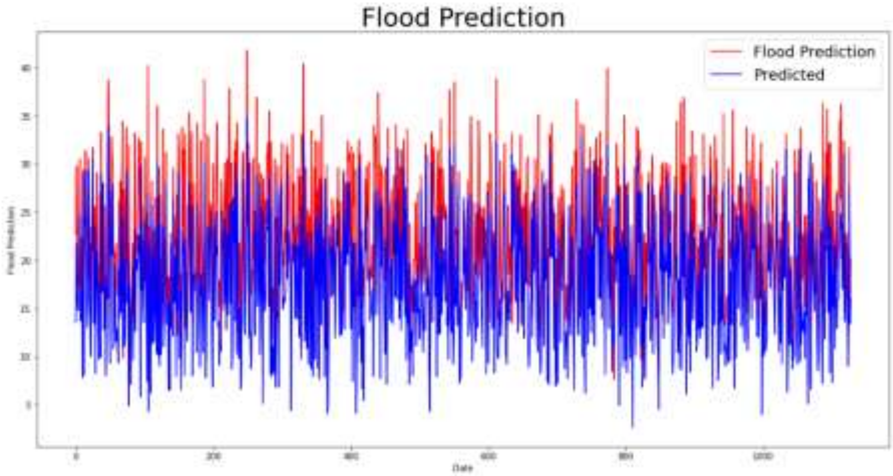

Figure 4: Time series visualization for the original Transformer model

This indicates that both actual results and predicted results are quite close. The increase and decrease in time series are consistent.

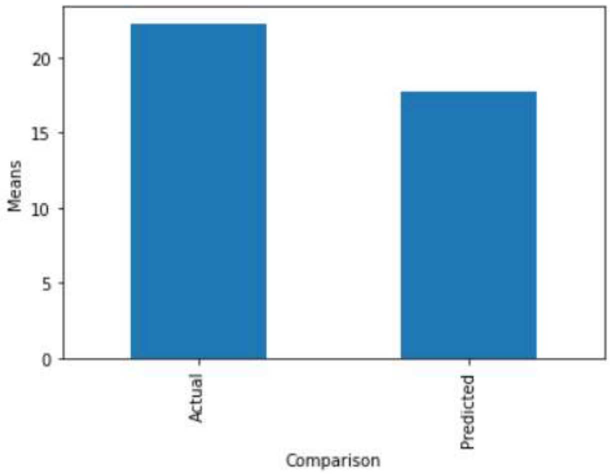

Figure 5: Comparison of Means for the original Transformer model

The above comparison indicates that the actual mean is higher than the predicted mean.

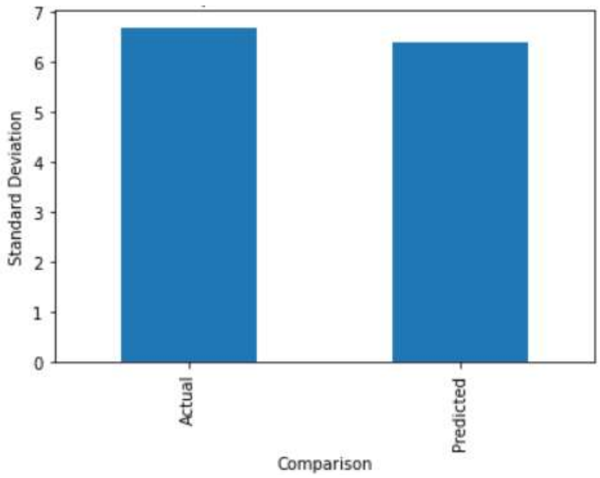

Figure 6: Comparison of Standard Deviation for the original Transformer model

The above plot shows that the actual standard deviation is quite close to the predicted standard deviation for this model.

Paper ID: SR211022155010 DOI: 10.21275/SR211022155010

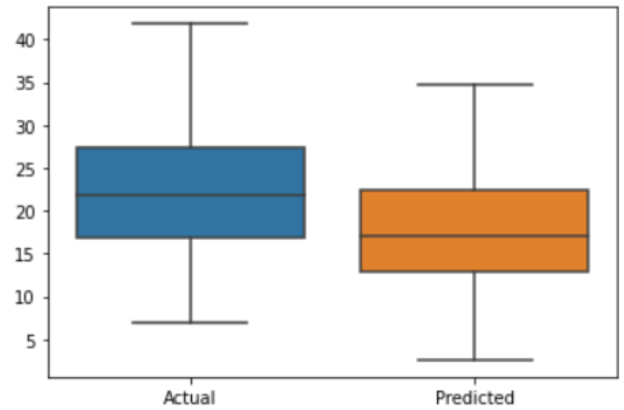

Figure 7: Interquartile Range for the original Transformer model

The overall results indicate that the Interquartile Range (IQR) is almost the same for both actual and predicted.

7. Comparison and Discussion

The performance of the transformer model is evaluated against other models. Not only does it compare time series datasets with alternative models, but it also tends to evaluate different configuration models in terms of hyperparameters. Different configurations tend to evaluate the dataset and its errors.

The original transformer models have a built-in activation layer. Here, the first model is a simple repetitive neural network (RNN) model, omitting the activation level. The second setup starts the model with 100 epochs, the third with 1000 epochs, and the newest model takes each day as a training set.

7.1 Model 1: RNN

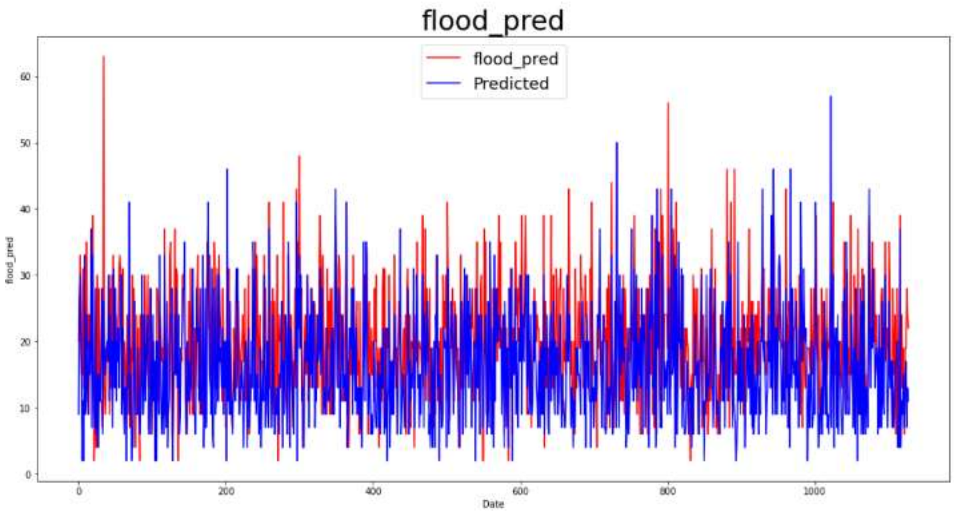

Figure 8: Time series visualization for Simple RNN model

In figure 8, the trend indicates closer estimates to the actual figure.

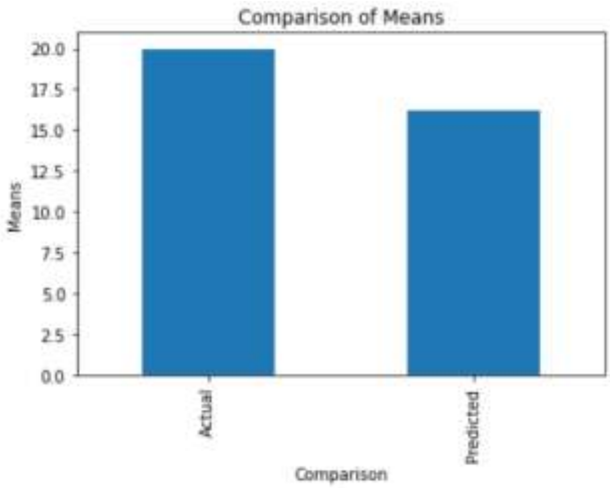

Figure 9: Comparison of Means for the Simple RNN model

Here, the actual mean and predicted mean are similar to the transformer model.

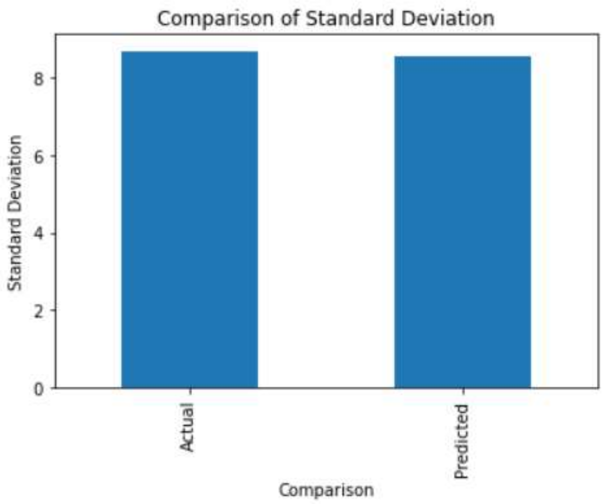

Figure 10: Comparison of Standard Deviation for the Simple RNN model

The standard deviations are also similar to the original transformer model.

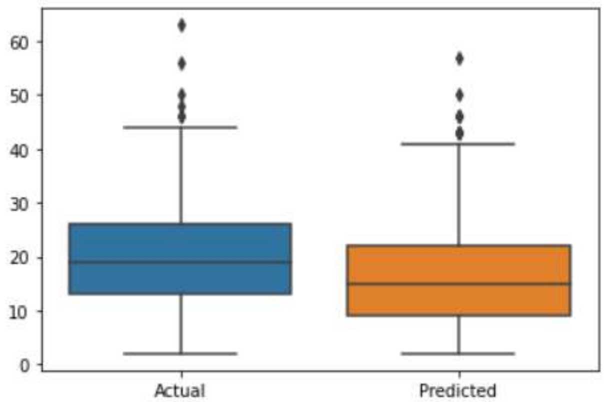

Figure 11: Interquartile Range for Simple RNN - Actual vs Predicted trend

In figure 11, the Box Plot illustrates that the trend of predictions is closer to the actual values.

7.2 Model 2: 100 Epochs

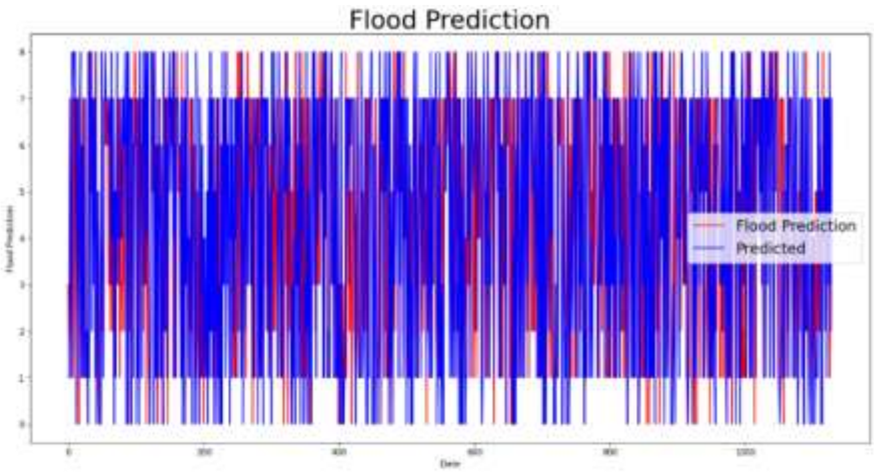

Figure 12: Time series visualization for a model trained on 100 epochs

In figure 12, the model indicates a significant gap between actual and predicted values.

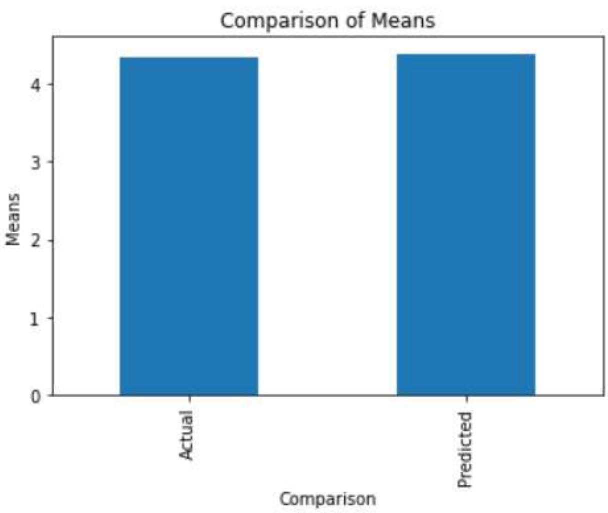

Figure 13: Comparison of Means for the model trained on 100 epochs

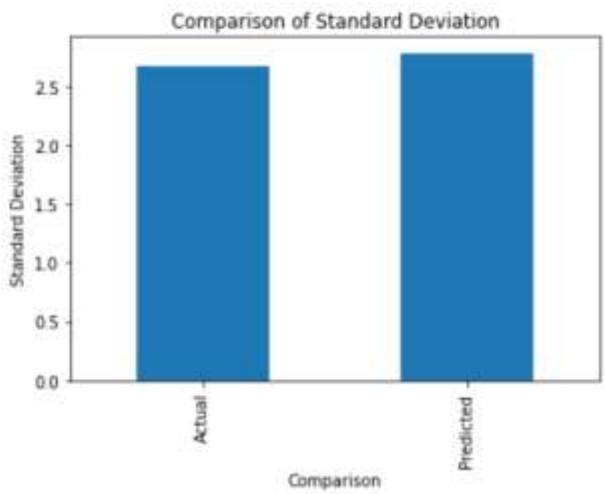

Figure 14: Comparison of Standard Deviation for the model trained on 100 epochs

The actual mean and the actual standard deviation are slightly higher than the corresponding predicted values.

Paper ID: SR211022155010 DOI: 10.21275/SR211022155010

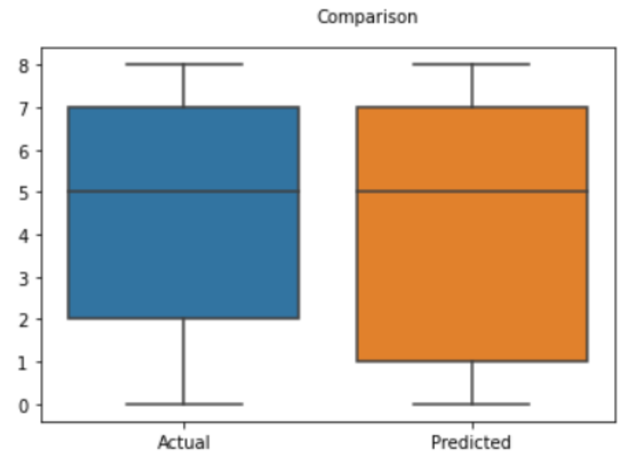

Figure 15: Interquartile Range for the model trained on 100 epochs

In figure 15, the result is in sharp contrast with the original transformer result.

7.3 Model 3: 1000 Epochs

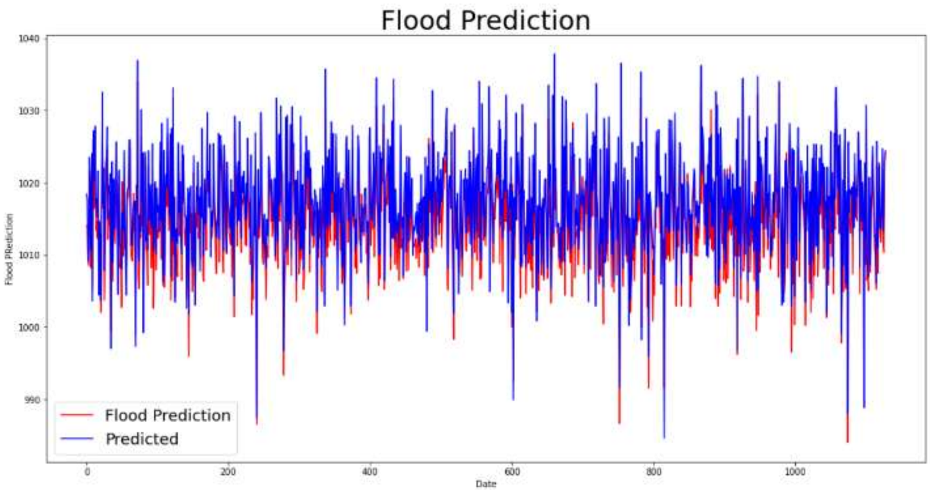

Figure 16: Time series visualization for the model trained on 1000 epochs

From figure 16, it can be discerned that this is better than the model trained on 100 epochs but not as accurate as the original model with 200 epochs.

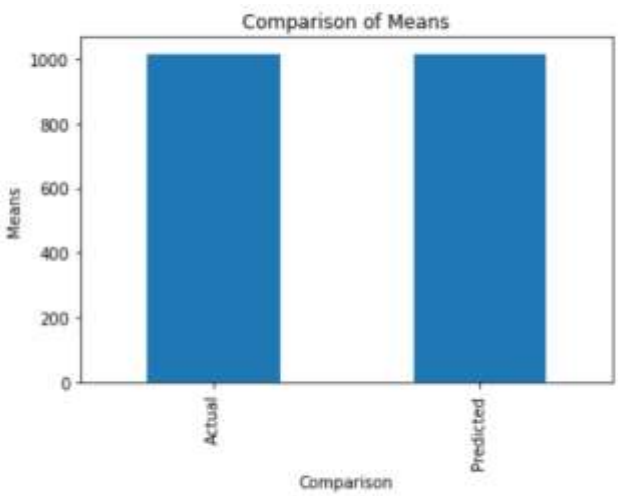

Figure 17: Comparison of Means for the model trained on 1000 epochs

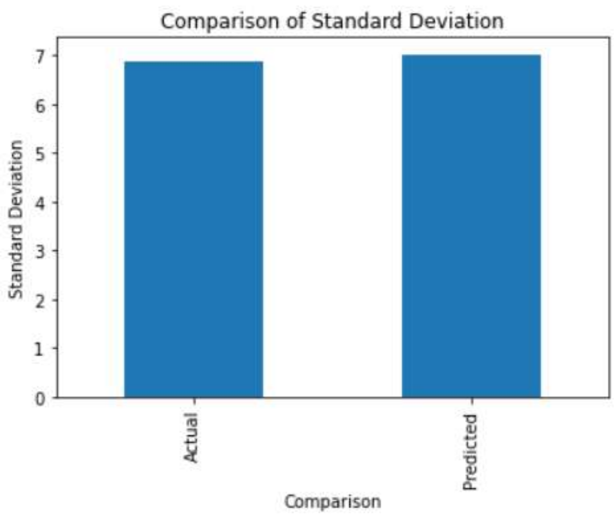

Figure 18: Comparison of Standard Deviation for the model trained on 1000 epochs

Here, the model indicates that the actual standard deviation is almost the same as the predicted standard deviation.

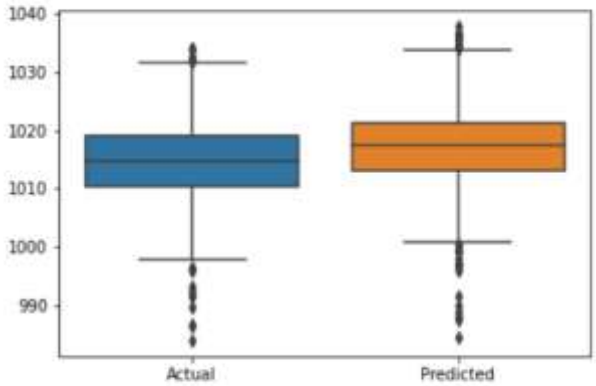

Figure 19: Interquartile Range for the model trained on 1000 epochs

The shape of both box plots is approximately the same, and it also indicates that the median and the maximum are higher for predicted.

7.4 Model 4: Alternative Day

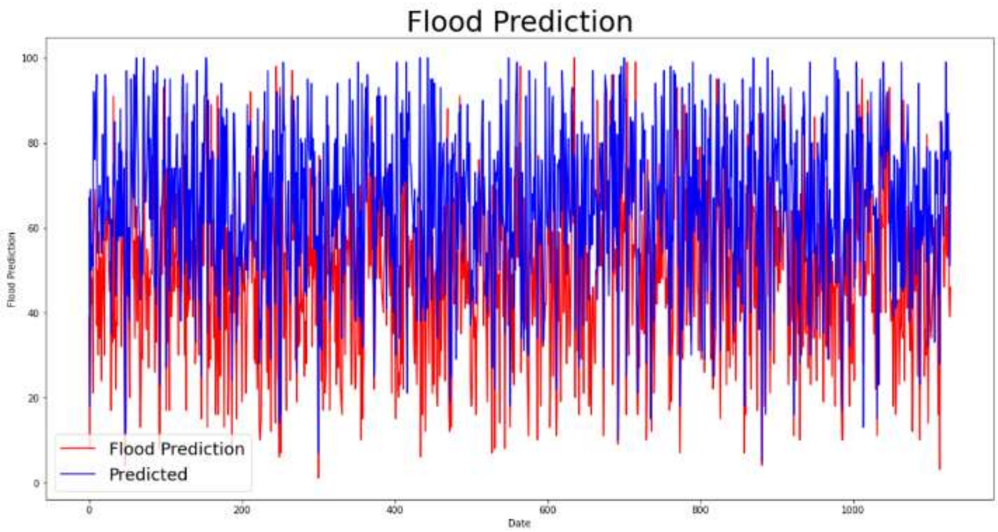

Figure 20: Time series visualization for the model trained on alternative days

In figure 20, the predicted results do not clearly depict the overall trend of the actual distribution.

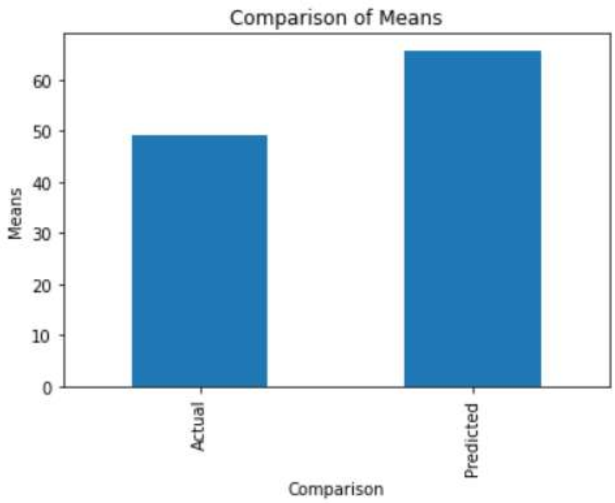

Figure 21: Comparison of Means for the model trained on alternative days

The predicted mean is higher, as the results show an average trend.

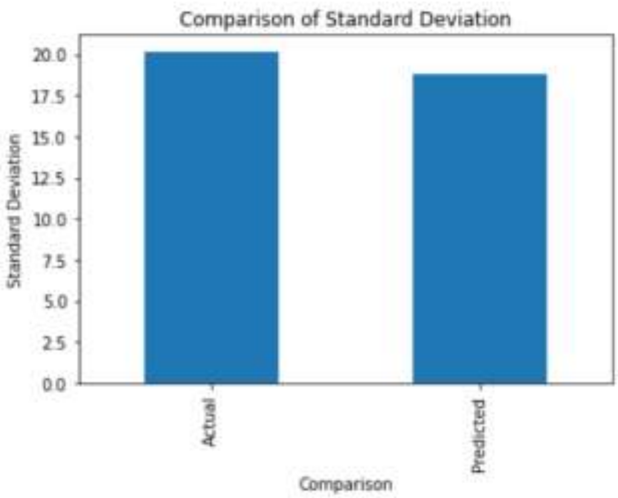

Figure 22: Comparison of Standard Deviation for the model trained on alternative days

This model has a predicted standard deviation lesser than the actual standard deviation.

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

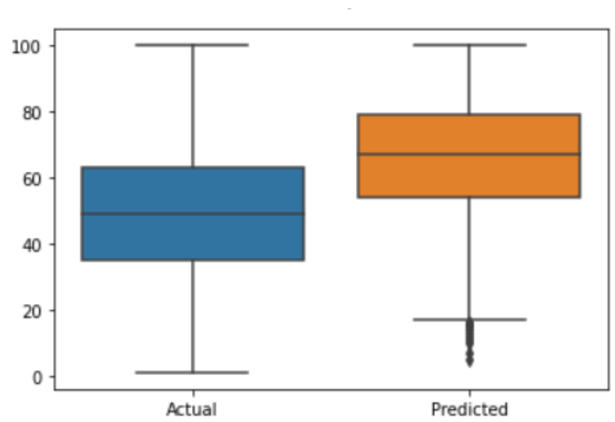

Figure 23: Interquartile Range for the model trained on alternative days

The overall predicted trend is better in the original transformer model.

8. Summary

Table 1: Standard measures deployed for the comparison of the different models

	Actual	Simple RNN	Epochs 100	Epochs 1000	Alt Days
Explained variance	-8.34155E-12	-9648.477254	-0.108572254	6.19171E-13	-1.17167E-07
Max error	0.01	97.55297985	0.999760027	1	0.999997734
Mean absolute error	0.012497694	12.69669504	0.150480201	0.124976941	0.124975153
Mean squared error	0.010818007	1054.369696	0.111598413	0.108180067	0.108179455
Mean squared log error	0.005277282	2.794904892	0.055759299	0.052772816	0.052772369
Median Absolute Error	0.001329402	2.061759311	0.028884304	0.013294015	0.013291372
Coefficient of Determination (R^2 score)	16.87456309	-11390.10018	-0.205676445	-0.168745631	-0.16873902
Mean Poisson Deviance	0.523677967	24.2405738	1.797628019	5.759595777	2.868821389
Mean Tweedie Deviance	0.010818007	1054.369696	0.111598413	0.108180067	0.108179455

9. Conclusion and Future Scope

The transformer model predicts flooding for the next day in Australia. Looking at the time-series graph, we can say that the transformer model has achieved promising accuracy. After comparing it with different models such as RNN (without activation layer), transformer model trained at 100 epochs and 1000 epochs, transformer model trained on alternate days, we conclude that the original transformer model is the best model for predicting time series.

References

[1] Box, G. E., Jenkins, G. M., & MacGregor, J. F. (1974). Some recent advances in forecasting and control. Journal of the Royal Statistical Society: Series C (Applied Statistics), 23(2), 158-179.

[2] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

[3] Choi, K., Hawthorne, C., Simon, I., Dinculescu, M., & Engel, J. (2020, November). Encoding musical style with transformer autoencoders. In International Conference on Machine Learning (pp. 1899-1908). PMLR.

[4] de Bézenac, E., Rangapuram, S. S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., ... & Januschowski, T. (2020, January). Normalizing Kalman Filters for Multivariate Time Series Analysis. In NeurIPS.

[5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[6] Du, S., Li, T., Yang, Y., & Horng, S. J. (2020). Multivariate time series forecasting via attention-based encoder–decoder framework. Neurocomputing, 388, 269-279.

[7] Kurle, R., Rangapuram, S. S., de Bézenac, E., Günnemann, S., & Gasthaus, J. (2020). Deep Rao-Blackwellised particle filters for time series forecasting. Advances in Neural Information Processing Systems, 33.

[8] Lai, G., Chang, W. C., Yang, Y., & Liu, H. (2018, June). Modeling long-and short-term temporal patterns with deep neural networks. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 95-104).

[9] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting.

[10] Maddix, D. C., Wang, Y., & Smola, A. (2018). Deep factors with gaussian processes for forecasting. arXiv preprint arXiv:1812.00098.

[11] Mohammdi Farsani, R., & Pazouki, E. (2021). A transformer self-attention model for time series forecasting. Journal of Electrical and Computer Engineering Innovations (JECEI), 9(1), 1-10.

[12] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 36(3), 1181-1191.

[13] Sen, R., Yu, H. F., & Dhillon, I. (2019). Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. arXiv

preprint arXiv:1905.03806.
[14] Shih, S. Y., Sun, F. K., & Lee, H. Y. (2019). Temporal pattern attention for multivariate time series forecasting. Machine Learning, 108(8), 1421-1441.
[15] Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., & Lendasse, A. (2007). Methodology for long-term prediction of time series. Neurocomputing, 70(16-18), 2861-2869.
[16] Sønderby, C. K., Espeholt, L., Heek, J., Dehghani, M., Oliver, A., Salimans, T., ... & Kalchbrenner, N. (2020). Metnet: A neural weather model for precipitation forecasting. arXiv preprint arXiv:2003.12140.
[17] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. arXiv preprint arXiv:2106.13008.
[18] Wu, N., Green, B., Ben, X., & O'Banion, S. (2020). Deep transformer models for time series forecasting: The influenza prevalence case. arXiv preprint arXiv:2001.08317.

Authors Profile

Mahek Jain, is currently a final year undergraduate student in the Computer Science and Engineering department at Rashtreeya Vidyalaya College of Engineering, Bangalore, Karnataka, India. She is an IT head and member of Ashwa Racing student organization at RVCE, which builds racing cars and competes at global level competitions. Her current research interests include applying deep learning algorithms in autonomous vehicle modules for pattern recognition and computer vision problems.

Shivam Pandey, is currently a graduate student in the Computer Science department at University of California, Davis, USA. He has secured his B.Tech degree from Indian Institute of Technology (IIT), Goa. He is an IEEE Professional Member and ACM Student Member. His current research area includes quantum machine learning and scientific computing.