2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) | 979-8-3503-4846-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/SCEECS61402.2024.10482217

Transformer-Based Multivariate Time Series Forecasting

1st Mukesh Kumar Bharti
Department of CSE
Maulana Azad National Institute of Technology
Bhopal, India
0009-0000-4073-1463

2ndDr. Rajesh Wadhvani
Department of CSE
Maulana Azad National Institute of Technology
Bhopal, India
0000-0002-2048-997X

3rd Dr. Manasi Gyanchandani
Department of CSE
Maulana Azad National Institute of Technology
Bhopal, India
0000-0003-3127-1770

4th Muktesh Gupta
Department of CSE
Maulana Azad National Institute of Technology
Bhopal, India
0000-0002-3135-8588

***Abstract*—This study explores multivariate time series forecasting, centering on the transformer model. It examines the shortcomings of other predictive models like Recurrent Neural Networks (RNN) and Temporal Convolutional Networks (TCN), particularly their inadequacies in handling autocorrelation. The transformer model stands out for its accuracy, thanks to its attention mechanism that focuses on essential parts of the input. The research introduces a novel approach that employs the transformer's architecture for effective feature selection in time series data. A vital aspect of this approach is the use of unsupervised pre-training, which shows superior results compared to traditional fully supervised methods. This advancement underscores the effectiveness of unsupervised learning in time series regression, offering significant benefits for diverse scientific and industrial fields.**

***Index Terms*—Time Series Forecasting, Deep Learning, Transformers, Encoder-Decoder, Attention Mechanism, Autocorrelation, Quantile Regression, Heteroskedasticity.**

I. Introduction

In time series forecasting [1]–[6], future values are predicted based on past or observed data. This research delves into the realm of multivariate time series forecasting, employing sophisticated techniques and methodologies. A weather forecast application is used as a case study, aiming to predict various meteorological elements such as precipitation, temperature, humidity, and wind speed. Weather forecasting is necessary because every year, any state is affected by a flood in India or any other country. Many people's houses sank due to the flood and many people died, the real example of this incident is of Kedarnath flood tragedy which happened in 2013. By doing weather forecasts we can save many people's lives from floods. Weather forecasting faces challenges due to the complex nature of atmospheric processes, including the chaotic behaviour of small-scale phenomena and the limited observational data available for accurate predictions.

Weather forecasting data represents a quintessentially complex and nonlinear domain, characterized by intricate patterns and unpredictable variables. The complexity arises from the myriad of interrelated factors that influence weather conditions, such as atmospheric pressure, temperature, humidity, and wind patterns. Each of these elements interacts in a highly nonlinear manner, meaning that small changes in one factor can lead to significant, often disproportionate, effects on the overall weather system. This nonlinearity is further compounded by the chaotic nature of the Earth's atmosphere, where variables are sensitive to initial conditions, making precise predictions challenging. Additionally, external influences like topographical features and ocean currents introduce further layers of complexity. As a result, accurately forecasting weather demands advanced computational models that can handle this nonlinearity and complexity, underscoring the need for sophisticated approaches in meteorological science. This complexity not only makes weather forecasting a challenging task but also a critical one, as accurate predictions are essential for planning and preparedness in various sectors, ranging from agriculture to disaster management therefore the traditional models like the Autoregressive Integrated Moving Average (ARIMA) [5] model, Exponential Smoothing (ETS) [6] methods, Seasonal Decomposition of Time Series (STL) [7] and other models are not able to handle the long-term dependencies and the traditional model could not be able to predict better as compare to the new deep learning model like RNN [8], TCN [9] and Transformer model [10]. Due to the complex and large dataset, the RNN model is not able to predict better and it takes much time to process because the RNN model processes input sequences by considering each element in the sequence one at a time and the TCN model also does not predict better compared to the transformer model because TCN models may face challenges in capturing very long-range dependencies and non-sequential patterns compared to Transformer model, which inherently excels at handling global contexts and complex relationships in sequential data [4]-[11]. The fixed receptive field of TCNs

979-8-3503-4846-0/24/$31.00 ©2024 IEEE

and reliance on convolutional operations contribute to these limitations. Applying a Transformer model to weather datasets for time series forecasting may encounter challenges due to the inherent non-uniformity of temporal patterns in weather data, requiring careful handling of irregular time intervals. Additionally, the substantial amount of missing or incomplete observational data in weather datasets poses a challenge, as the Transformer model typically assumes complete sequences for optimal performance.

The objectives are to prepare time series data by normalizing, removing outliers, and addressing missing values. Enhance model efficiency by extracting key features through techniques like feature selection or dimensionality reduction. Optimize model parameters to prevent overfitting and ensure generalization, while implementing parallel computing strategies for faster training and inference. In this paper, we used the Transformer model because, in recent times, the Transformer model has become widely favoured for its superior performance in multivariate time series forecasting when contrasted with alternative models.

The key contributions of this research are outlined as follows:

- Data preprocessing was performed on the Weather dataset.
- Feature selection techniques were applied.
- Utilized a transformer model with a self-attention mechanism.
- Achieved superior results compared to RNN and TCN models and other models.

TABLE I
LIST OF ABBREVIATIONS

Abbreviations	***Full Form***
ARIMA	AutoRegressive Integrated Moving Average
SARIMA	Seasonal ARIMA
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
TCN	Temporal Convolutional Network
FNN	Feedforward Neural Network
ST.D.	Standard Deviation

II. Background

A. *Time Series Forecasting*

Time series forecasting refers to the forecasting of future values based on observational data or past data. For example, in the case of weather forecasting predicting the temperature of the next day based on past data which can be from many years past data and based on these data, we can predict the temperature, precipitation, humidity, wind speed and many more. There are two types of time series forecasting one is univariate time series and another is multivarite time series forecasting. In univariate time series forecasting, there are single variables or observations recorded at different time points. For example, The forecast for future daily temperatures at a specific location relies exclusively on historical temperature records, with the primary variable being the daily temperature recorded at various time points. In multivariate time series forecasting, there are multiple variables or observational data are recorded at each time step. For instance, forecasting future daily conditions may consider variables such as temperature, humidity, precipitation, and wind speed, recorded at various time points. This approach allows for a more comprehensive prediction by incorporating the interdependencies among multiple weather-related factors over time. Given the recent improvements in deep learning-based frameworks, transformers are demonstrating promising results, suggesting an avenue for further study. This section will focus on attention-based methodologies already employed by several writers in time series forecasting.

B. *Related Works and Literature Survey*

Nowadays, Deep learning models become an important model in time series forecasting due to their capacity to recognize intricate data patterns. The Deep learning models like RNN and LSTM [13] are popular deep learning techniques because of their ability to simulate temporal correlations in time series data. The Convolutional neural network(CNN) [14] can also be employed in time series forecasting by treating the time series as an image. The Gated recurrent units(GRU) [15] can also be used to predict future values. GRU addresses some of the issues like the vanishing gradient problem and difficulties in capturing the long-term dependencies and these issues belong to the RNN model. However there are some limitations of the GRU model also, GRUs might not perform as well when the time series exhibits intricate patterns or requires a nuanced understanding of context, and fine-tuning hyperparameters can be challenging. While achieving superior outcomes and demonstrating exceptional efficacy across diverse applications, these deep learning algorithms excelled. In recent years, the transformer model, leveraging the attention mechanism, has become popular in time series forecasting, diverging from its primary use in natural language processing (NLP) [10].

The paper on Deep Adaptive Input Normalization [2], authored by Nikolaos Passalis et al., explores learning data normalization and adaptive normalization scheme adjustments during inference using the deep adaptive input normalization layer. The study employs a deep learning model, such as RNN, facing challenges in capturing long-term dependencies, necessitating updates to the parameters of the normalization layer. Additionally, the proposed scheme covers special cases like mean normalization, z-score, and min-max normalization, with a specific focus on a stock prediction dataset featuring attributes like historical open, close, low, and high values or prices.

The Deep Transformer Model for Time Series Forecasting paper [11] predicts illnesses using reports from the Centers for Disease Control and Prevention (CDC) in the United States. It employs a transformer model and a deep learning approach based on RNN to model illness data, yet faces challenges capturing long-term and complex relations in sequence data due to issues like exploding and vanishing gradients. The dataset includes attributes like date and time, body mass index, medication records, age, gender, and symptom records. In a study of a comparable nature conducted by Shengdong Du et al. [16], the transformer-based network employs the attention mechanism used in multivariate time series forecasting. They worked on various datasets, including PeMS Bay, Highway traffic, Italian air quality, and Beijing PM25 to demonstrate the effectiveness of their findings. The RMSE and MAE values obtained higher across the board. They compared the result of the several models to their MTSMFF model, the several models including ARIMA, RNN, LSTM, CNN, and Sequence-to-Sequence model.

C. *Research Gaps*

After surveying the research paper we identified several research gaps. One is Dealing with high-dimensional data. While deep learning has shown good performance in many time series forecasting tasks, they may need help dealing with high-dimensional data with very large features. Second is Robustness to autocorrelation; time series data often contains outliers and anomalies that can significantly impact the accuracy of forecasting models. Existing models for time series don't effectively handle autocorrelation data. Existing time series models are not effective in capturing long-term dependencies. Third, many times, series forecasting models are developed for specific applications or datasets and may not generalize well to new datasets or applications. Research is needed to develop models that can generalize across different time series datasets and applications.

III. Proposed Methodology

Current research highlighted in the preceding section demonstrates the multivariate time series forecasting based on the transformer model, which produces better results. Using data preprocessing to handle the outliers, fill in the missing values, remove duplicates, and many more. Next, use of feature selection techniques for wrappers, filters, or positional embedding; the most relevant properties can be retrieved from the multivariate dataset. In the next step, in the transformer model, these chosen qualities are provided so that it can explain the complex relationships between the variables accurately and capture their temporal dependencies. For the time series prediction problem, the transformer model, which uses the attention mechanism, can assist in determining the most important components and the time steps. Utilizing both feature selection and transformers can enhance the efficiency of the model, mitigate overfitting, and improve its ability to predict a multitude of interconnected variables over time accurately. We employed comparable procedures on our model, incorporating pre-processing techniques like data cleansing, scaling, data imputation, and normalization (as shown in Fig.1). This was done to ensure that the data meets the necessary standards for utilization by our model.

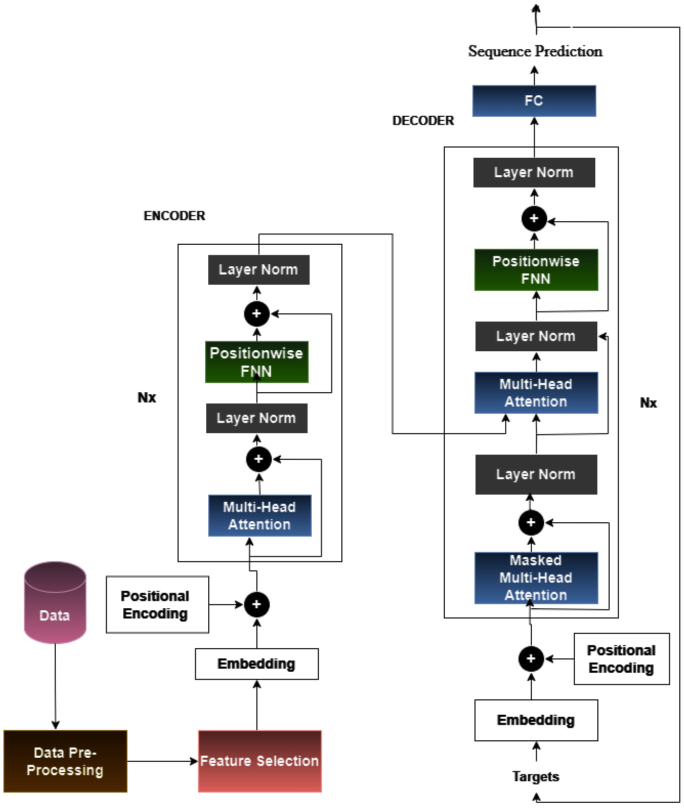

Fig. 1. Proposed Architecture ('Nx' represents the repetitions of Encoder and Decoder).

A. *Feature Selection*

In many machine learning including deep learning, feature selection [12] is the initial and important stage to choose the most important and relevant characteristics. The most important is that it can reduce the dimensionality of the input space, accelerate training durations, and increase the model performance of deep learning. Feature selection is essential to enhance accuracy, as raw data without this technique results in lower accuracy output. There are three methods involved in the feature selection method. One is the filter method in which we choose the relevant attribute to the target value. In the Wrapper method, attributes or combinations are selected and used in training to generate multiple models, and their accuracy is assessed. Third is the Embedded method in which we select useful attributes, this method reduces overfitting problems compared to the wrapper method. Feature selection is crucial in deep learning as it helps in identifying the most relevant information within a dataset.

B. *Positional Encoding*

The computer does not understand the word, so it needs to convert the words into numbers or vectors in matrices [10]. Utilizing an embedding space translates a word into a vector; however, when the same word appears in distinct sentences with varied meanings, positional encoders become crucial.

These encoders contain a vector that conveys information about the spatial gaps between words within the sentence. The original paper uses sin and cos functions to generate the vector but it could be any reasonal function.

Here is the formula to calculate the positional encoding:

$$\text{PE}(p, 2j) = \sin\left(\frac{p}{10000^{2j/d}}\right) \tag{1}$$

$$\text{PE}(p, 2j+1) = \cos\left(\frac{p}{10000^{2j/d}}\right) \tag{2}$$

Where 'j' is the dimension index, 'p' is the element's position in the sequence, and 'd' is the embedding's dimensionality.

The transformer model analyses the input sequence in parallel and needs an explicit method to incorporate positional information. While the recurrent neural network processes one word at a time, the longer input takes longer, and RNN could not be able to understand the word context. The Positional encoding can understand the order of data points.

C. *Encoder*

The Encoder is the crucial component in the transformer architecture, which utilizes natural language processing and other sequence-to-sequence processes. The encoder captures temporal patterns and relationships within the sequential data for effective representation learning. The transformer model uses multiple encoder layers and each layer consists of two sublayers one is position-wise fully connected feed-forward networks and the second is a multihead self-attention mechanism [10]. A feed-forward layer is essential for introducing non-linear transformations, enabling the model to grasp intricate patterns and relationships within the input data, thereby enhancing its capacity for learning and generalization from sequential information. For each input point, the encoder produces the fixed-size representations. Within the Transformer architecture, the Encoder captures essential input sequence information and transforms it for the Decoder's utilization.

D. *Attention Mechanism*

The attention mechanism [10] employs a strategy enabling a neural network to concentrate on specific segments within the input sequence. In this weights are assigned to the different sections of the input sequence, with the most crucial segment receiving the highest weights. In contrast to conventional models like RNN, which process one word at a time, an attention model differs in that the encoder transmits a more extensive set of data to the decoder. Rather than forwarding only the final hidden state, the encoder conveys all hidden states from each time step to the decoder. The attention mechanism is to turn the information into the vector of query, values, and keys. The query represents the input seeking attention, the key signifies elements determining attention, and the values are associated with information related to each key. The mechanism calculates similarity to weight values, producing a weighted sum for further computations. Based on the current query, the self-attention method enables the model to dynamically assign weights to different segments of the input sequence.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

Here, the key vector is represented by 'K', the query vector is represented by 'Q', the values vector is represented by 'V', and the dimension is represented by 'd'.

E. *Decoder*

From the Encoder's encoded representation, the Decoder in the transformer architecture generates the output sequences. The decoder in the transformer architecture contains three sublayers which are positional-wise feed forward networks, attention mechanism, and masked multihead attention mechanism. Due to the multi-head self-attention system, the decoder can look back. Each decoder position's hidden state can be transformed by the positional-wise feed-forward network that would be non-linear.

F. *Softmax*

The transformer model uses the softmax layer, where softmax is the activation function that is commonly applied to normalize attention scores during the attention mechanism. This normalization ensures that the model assigns appropriate weights to different elements in the sequence. The softmax layer helps in capturing the significance of each time step, contributing to more accurate predictions in time series forecasting.

$$\text{softmax}(a)_i = \frac{e^{a_i}}{\sum_{j=1}^{n} e^{a_j}} \tag{4}$$

Here, the input value is represented by 'a_i' for the 'i^{th}' element and 'n' total number of elements for the vector or sequence.

IV. Experiments

To evaluate the transformer model which we used in this paper, we did several different tests such as ablations, comparison, and in-depth study. we will elucidate the findings of the performance outcomes and provide details regarding the prediction experiment in this section of the report.

A. *Dataset*

The multivariate weather dataset [17] has been used to evaluate the proposed model. The dataset is sourced from the US government website and relates to the Bairagarh Airport, Bhopal, India. In terms of data volume, this dataset is quite substantial, comprising a total of 13,318 rows or data points. It contains the daily measurement of a duration of 37 years from 1973 to 2010, capturing daily measurements.

Various weather attributes (4 features) are recorded, including temperature, humidity, and precipitation. All data entries are synchronized to a consistent time axis to simplify usage. due to the data cleaning procedure and measurement error, some data are missing. The weather descriptions are given in Table II.

TABLE II
WEATHER DATASET DESCRIPTION [17]

	Weather Dataset			
	PRCP	***TAVG***	***TMAX***	***TMIN***
Mean	0.18	25.37	31.94	18.76
St.d.	0.68	5.21	5.55	5.77
Min	0.00	9.44	13.33	0.00
Max	18.54	39.44	47.22	32.78

B. *Data Pre-Processing*

For the analysis, Machine learning and data science involve cleaning and transforming the unprocessed data. Data preprocessing is the crucial stage, the model can produce better accuracy by doing data preprocessing compared to the unprocessed data. Data preprocessing improves the quality of data, increases the efficiency, and ease of the mining process, and removes noisy data, incomplete data, and inconsistent data.

Time series forecasting uses several stages for data preprocessing. First is data cleaning in which data are cleaned by the missing values by backfill or forwardfill or interpolate method, smoothing the noisy data, removing the outliers, and resolving the inconsistency. Second, is data transformations in which we normalise or scale the feature, encode the categorical variables, and handle the outliers from the dataset. Third is data reduction which uses techniques like feature selection or principle component analysis to reduce the dimensionality of the dataset, and then sampling or aggregating data for efficiency. In addition, other techniques such as data integration and data discretization are also used. Missing data may lead to biased outcomes, reduced statistical power and sample size, and challenges in visualizing and summarizing data.

There are outliers observed in the dataset and there is heteroscedastic distribution and autocorrelation in the data. The transformer model has the ability to resolve the heteroscedastic and autocorrelation problems. To manage the outliers the z-score [18] was used. Our experimentation involved both KNN Imputer [19] and Iterative imputation [20]. Given that the outcomes from KNN imputation did not align with our results, we proceeded to explore an alternative method. During the feature engineering process, we include additional data features like week of the year, week of the month, and day of the week to prevent any loss of crucial relationships.

C. *Hyper Parameter Details and Tuning*

There are several parameters of the transformer model that can be fine-tuned through experimental analysis. The number of layers in each encoder and decoder may be adjusted based on the input sequence. The experiments were done by the various input chunk lengths, counts of multi-head attention, batch size, activation functions such as ReLU [21] and GELU [22], and expected feature counts that a transformer model anticipates. Likewise, in the process of feature selection, various approaches, such as the Pearson correlation coefficient [23] and Principal Component Analysis (PCA) [24], were examined at different association levels. This exploration aimed to assess the correlation between the independent and dependent variables within the datasets. There are 4 encoders-decoders used which is less than the original transformer model, the dropout was set to 0.1, ReLU was used as an activation function, the batch size was 32, the headcount was 8, the learning rate was set to 0.0001, and the input chunk length was 64. This configuration improved the performance of the model.

D. *Performance Metrics*

For measuring the performance of the proposed model there are two most important techniques RMSE and MAPE are used.

RMSE(Root Mean Squared Error): RMSE serves as an indicator of the typical disparity between forecasted and observed values for a variable, considering the scale of the discrepancies. This measure is particularly beneficial in scenarios involving normally distributed errors and variables of a continuous nature.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - \hat{p}_i)^2} \tag{5}$$

Here, actual values are represented by 'a_i', the predicted values are represented by '$\hat{p}_i$' of the variables at the 'i^{th}' observation and 'n' total recorded data in the dataset.

MAPE(Mean Absolute Percentage Error): MAPE gauges the mean percentage distinction between predicted and actual values for a variable. This metric proves valuable in situations involving variables with diverse scales or units. Here is the formula for MAPE calculation:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{a_i - \hat{p}_i}{a_i}\right| \times 100 \tag{6}$$

Here, the actual values are represented by 'a_i' of the variable and the predicted values are represented by '$\hat{p}_i$' of the variables at the 'i^{th}' observations and 'n' total recorded data in the dataset.

E. *Results and Discussion*

The weather feature dataset [17] has been extensively tested using established techniques like RNN, LSTM, GRU, and TCN to evaluate the performance of the proposed model. This dataset was applied not only to these traditional models but also to our novel model for comparative analysis. Our findings indicate that the proposed model surpasses these existing models in terms of accuracy. Fig.3, shows how accurately the weather prediction of temperature matches with the actual values. 10% of the data was used for testing and 90% data was used for training. Table III shows the comparisons of the traditional models and the proposed model, with the help of two important techniques RMSE and MAPE. The less the RMSE more will be the accuracy. In the above table, the proposed transformer model beats all the other traditional models.

TABLE III
MULTIVARIATE TIME SERIES ANALYSIS ON WEATHER DATASET [17].

Model	Weather Dataset		
	RMSE	*MAPE*	*Accuracy (in %)*
RNN	3.00	8.43	87.27
LSTM	4.48	9.29	82.62
TCN	5.85	18.51	77.17
Transformer (Proposed Model)	**2.89**	**8.80**	**88.70**

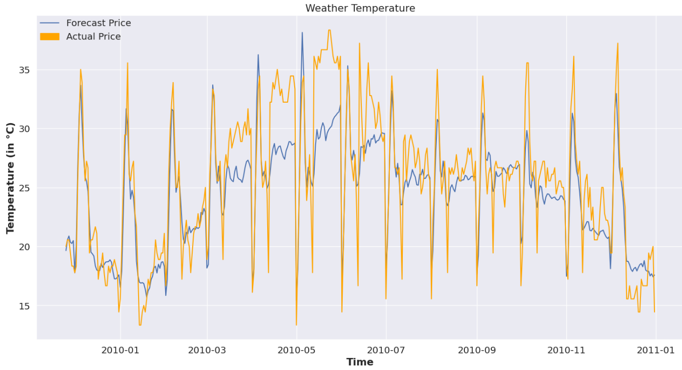

Fig. 2. The prediction graph of the RNN model on the Weather dataset [17].

V. Conclusion

In this study, we introduced an innovative approach for forecasting multivariate time series, combining the process of selecting features with the construction of a transformer model. Through analysing the correlation between input factors and temporal characteristics, our approach aims to precisely forecast a diverse range of variables over extended durations. After the study of the literature survey, we found the deficiencies that need to be worked on. We used the proposed model on the weather feature dataset that produced the state-of-the-art prediction compared to the traditional model.

References

[1] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, *”Transformers in time series: A survey,”* arXiv preprint arXiv:2202.07125,(2022)

[2] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj and A. Iosifidis, *”Deep Adaptive Input Normalization for Time Series Forecasting,” in IEEE Transactions on Neural Networks and Learning Systems”*, vol. 31, no. 9, pp. 3760-3765, doi: 10.1109/TNNLS.2019.2944933, (Sept. 2019).

[3] D. Patil, R. Wadhvani, S. Shukla, and M. Gupta, *“Adaptive wind data normalization to improve the performance of forecasting models,” Wind Engineering,* vol. 46, no. 5, pp. 1606–1617, (2022).

[4] J. Li, X. Han, and S. Li, *“Transformer-based time series imputation and prediction for heteroscedastic data,”* IEEE Access, vol. 9, pp. 146685–146696, (2021).

[5] M. Valipour, *“Long-term runoff study using sarima and arima models in The United States,” Meteorological Applications,”* vol. 22, no. 3, pp. 592–598, (2015).

[6] Shastri, Sourabh Sharma, Amardeep Mansotra, Vibhakar Sharma, Anand Bhadwal, Arun Kumari, Monika. *”A Study on Exponential Smoothing Method for Forecasting.”* International Journal of Computer Sciences and Engineering. 6. 482-485. 10.26438/ijcse/v6i4.482485, (2018).

[7] Dagum, Estela. *”Time Series Modelling and Decomposition.”* Statistical. 70. 10.6092/issn.1973-2201/3597, (2013).

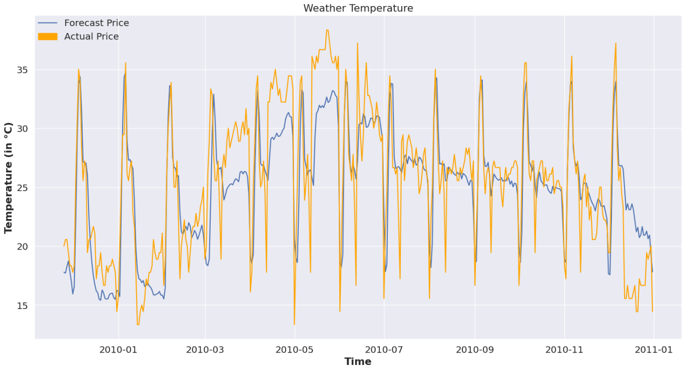

Fig. 3. The prediction graph of the Transformer model on the Weather dataset [17].

[8] W. Yin, K. Kann, M. Yu, and H. Schutze, *“Comparative study of cnn and ¨rnn for natural language processing,”* arXiv preprint arXiv:1702.01923, (2017).

[9] Bednarski, B.P., Singh, A.D., Zhang, W. et al. *”Temporal convolutional networks and data rebalancing for clinical length of stay and mortality prediction.”* Sci Rep 12, 21247 (2022).

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. *”Attention is All You Need. In Advances in Neural Information Processing Systems”. (pp. 5998-6008)*, (2017).

[11] Wu, Neo, et al. *”Deep transformer models for time series forecasting: The influenza prevalence case.”* arXiv preprint arXiv:2001.08317, (2020).

[12] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, *“Feature selection: A data perspective,”* ACM computing surveys (CSUR), vol. 50, no. 6, pp. 1–45, (2017).

[13] R. C. Staudemeyer and E. R. Morris, *“Understanding lstm–a tutorial into long short-term memory recurrent neural networks,”* arXiv preprint arXiv:1909.09586, (2019).

[14] L. O. Chua and T. Roska, *“The cnn paradigm,” IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,”* vol. 40, no. 3, pp. 147–156, (1993).

[15] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.”* arXivpreprint arXiv:1412.3555, (2014).

[16] S. Du, T. Li, Y. Yang, and S.-J. Horng, *“Multivariate time series forecasting via attention-based encoder-decoder framework,”* Neurocomputing, vol. 388, pp. 269–279, (2020).

[17] https://www.ncdc.noaa.gov/cdo-web/datasets/GHCND/locations/CITY:IN000006/detail

[18] S. Seo, *”A review and comparison of methods for detecting outliers in univariate data sets.”* PhD thesis, University of Pittsburgh, (2006).

[19] S. Zhang, *“Nearest neighbour selection for iteratively knn imputation,”*Journal of Systems and Software, vol. 85, no. 11, pp. 2541–2552, (2012).

[20] T. Klomp, *“Iterative imputation in python: A study on the performance of the package iterativeimputer,”* Master’s thesis, (2022).

[21] Agarap, A. F. (2018). *”Deep Learning using Rectified Linear Units (ReLU).”* ArXiv. /abs/1803.08375

[22] Lee, M. *”GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance.”* ArXiv. /abs/2305.12073, (2023).

[23] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, *“Pearson correlation coefficient,”* Noise reduction in speech processing, pp. 1–4, 2009.

[24] A. Gonzalez-Vidal, F. Jimenez, and A. F. Gomez-Skarmeta, *“A methodology for energy multivariate time series forecasting in smart buildings based on feature selection,”* Energy and Buildings, vol. 196, pp. 71–82, (2019).

[25] X. Liao, S. Hong, H. Yan, Y. Chang, Y. Cheng, W. Sun, Y. Li, Q. Liu, X. He, X. Fan, et al., *“UCI machine learning repository,”* (2017).