ConTra : Convolutional Transformer for Multivariate Long Sequence Time Series Forecasting

Akash M. Jadhav[*]
akashj0305@gmail.com

Siddharth M. Jadhav
siddharthj0305@gmail.com

Ritesh D. Nikose
ritesh.nikose23@gmail.com

Manish S. Doiphode
manishdoi08@gmail.com

Abstract

Accurate long-term forecasting of phenomena that are highly dependent on multiple factors remains a significant challenge due to its multivariate nature and the inability of the models to consider these factors. For example, long-term weather forecasting is a challenge in various sectors such as agriculture, disaster management, and urban planning where its dependencies on various meteorological factors must be considered. Traditional statistical and learning methods such as ARIMA and LSTM struggle to capture long-range dependencies, leading to limited predictive accuracy over extended periods. This paper introduces a hybrid model that integrates Convolutional Neural Networks(CNNs) with Transformer architectures named ConTra to enhance forecast performance for long-sequence time series data. The proposed hybrid CNN-Transformer model captures local patterns from the feature lattice using CNN's together with an advanced positional encoder to capture positional information for multivariate data, while the Transformer's attention mechanism handles long-range dependencies, effectively improving overall predictive accuracy. This study tested the proposed model on the Jena Climate dataset; the proposed model outperforms existing methods with an RMSE of 0.00799, MAE of 0.00163, MSE of 0.01762, and an R-squared value of 0.9991, demonstrating its potential to advance state-of-the-art weather prediction by providing reliable long-term forecasts.

Keywords: Long sequence forecasting, CNN, Transformer architecture, Jena Climate Dataset, Time series analysis, RMSE, MSE, MAE, R^2.

1 Introduction

Forecasting complex, multi-factor phenomena, especially over long sequences remains a challenge with critical applications in weather forecasting for agriculture, urban planning, and disaster preparedness. Seasonal Climate Prediction (SCP) is essential for understanding climate change impacts Doblas-Reyes et al. [2013]. Rising global temperatures and the increased frequency of extreme events, such as droughts and heatwaves, adds complexity and uncertainty to long-term forecasting.

Air temperature forecasting, a core aspect of SCP, involves highly complex and dependency on multivariate data, necessitating advancements in Machine Learning (ML) and Artificial Intelligence (AI) methods. Techniques such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM) have shown strong performance in climate data prediction Dombaycı and Gölcü [2009], Liu et al. [2012]. Enhanced accuracy has been achieved with advanced ML models like ANN optimized with the Levenberg–Marquardt(LM) algorithm, achieving high predictive success Kisi and Shiri [2014]. Recent deep learning models, including those with Stacked Denoising Autoencoders (SDAE)

[*]Corresponding author: akashj0305@gmail.com

1st Conference on Applied AI and Scientific Machine Learning (CASML 2024).

Hossain et al. [2015] and Long Short-Term Memory (LSTM) networks Li et al. [2019, 2020], have further improved temporal dependency capture in climate predictions.

Hybrid models, particularly those combining Convolutional Neural Networks (CNN) and LSTMs, enhance accuracy by integrating feature extraction and temporal dependency management Hou et al. [2022], Fister et al. [2023], Utku and Can [2023]. This study proposes a CNN-Transformer hybrid model, with CNNs for local feature extraction and Transformers for long-range dependencies, demonstrating superior performance on the Jena Climate dataset. Key metrics, including RMSE, MAE, MSE, and R^2, validate its advantages over current state-of-the-art methods.

Table 1: Comparison of various models for weather forecasting with respect to different performance metrics.

Model	**MSE**	**RMSE**	**MAE**	$\mathbf{R}^2$	**Reported by**
Levenberg–Marquardt	-	1.96550	-	0.98881	Dombaycı and Gölcü [2009]
Wavelet-SVM	0.0937	-	-	-	Liu et al. [2012]
Levenberg–Marquardt	-	1.53	1.27	0.995	Kisi and Shiri [2014]
SDAE	-	1.38	-	-	Hossain et al. [2015]
SVR, MLP	-	-	0.7232	-	Salcedo-Sanz et al. [2016]
Stacked-LSTM	1.5365	1.236	0.9056	0.9692	Li et al. [2019]
LSTM	-	1.04	-	0.984	Li et al. [2020]
CNN–LSTM	-	1.97	1.02	-	Hou et al. [2022]
Recurrence Plot+CNN +Binarised	0.718	-	0.696	-	Fister et al. [2023]
CNN-RNN	0.035	0.189	0.126	0.987	Utku and Can [2023]

2 Materials and Methods

2.1 Data set and prepossessing

The dataset used in this research is the *Jena Climate 2009–2016* Dataset [2022] dataset, which contains a continuous data of weather measurements recorded once every 10 minutes from the weather station at the Max Planck Institute for Biogeochemistry in Jena, Germany. The dataset spans from 2009 to 2016 and provides an extensive array of atmospheric measurements including temperature, pressure, humidity, wind speed, wind direction and different other measurements. These meteorological variables are essential for studying climate dynamics and performing weather prediction tasks, making this dataset an ideal candidate for various machine learning experiments, particularly for time-series analysis.

Each data point represents observations recorded once every 10 minutes over a seven-year period, resulting in a dataset of approximately 420,551 records. As per statistical distribution it was observed that there is no significant change within 60 min, thus the values were resampled to 60 min. One hundred twenty hours of data were tracked from the past 720 timestamps (720/6 = 120). This data was used to predict the temperature 24 h after 720 timestamps (720/6 = 120).The time-series nature of the data allows for the analysis of temporal trends and the development of predictive models for climate conditions. The dataset can be accessed through the *UCI Machine Learning Repository* under the *Jena Climate Dataset*.

Training, Validation, and Testing Data:
Following the preprocessing steps, the dataset was split into training, validation, and testing sets. The training set consisted of 70% of the data, while the validation and testing sets comprised 20% and 10%, respectively. This resulted in 48,654 sequences for training, 7,984 sequences for validation, and 6,828 sequences for testing, with each sequence containing 120 hours. These sequences were used to train and evaluate the model for 24 hours across the different phases.

2.2 Methodology

Long Sequence Time Series Forecasting (LSTF) requires models capable of capturing long-term dependencies and contextual information from sequences. Conventional models, such as ARIMA, LSTM, and LSTM-based variants, struggle to capture these dependencies effectively when sequences

are unusually long. Additionally, these sequential modeling methods are limited in their ability to leverage advances in parallel computing, leading to increased processing time and inefficiency.
While transformers, known for their efficiency in processing sequences, have become the computational backbone for large language models, they face limitations when applied to LSTF tasks. The primary challenge stems from their quadratic time complexity, which becomes problematic when handling multivariate time sequences.

Moreover, the transformer's positional encoding mechanism is inherently designed for univariate sequences. In this architecture, positional embeddings are generated across the model's depth, preserving positional information while allowing for parallel processing. However, this design presents a challenge for multivariate data, as each position in the sequence contains multiple values representing different features. Transformers, in their existing form, are not well-equipped to handle multivariate inputs due to this architectural constraint.

To address this limitation, proposed model leveraged the strengths of Convolutional Neural Networks (CNNs) by transforming the multivariate 1-D data into multiple 2-D layers. Each 2-D layer consisted of embeddings with a size equivalent to the model's depth (512), organized in a stacked fashion. As shown in 1, Positional information for these embeddings was stacked independently, maintaining the same stack size. These 2-D layers were then combined with the positional embedding stack by summing them together. The combined output was passed through convolution layers, where information from all stacked layers was consolidated into a single layer. This processed layer was then used as the input to the transformer's encoder, enabling it to effectively capture multivariate dependencies.

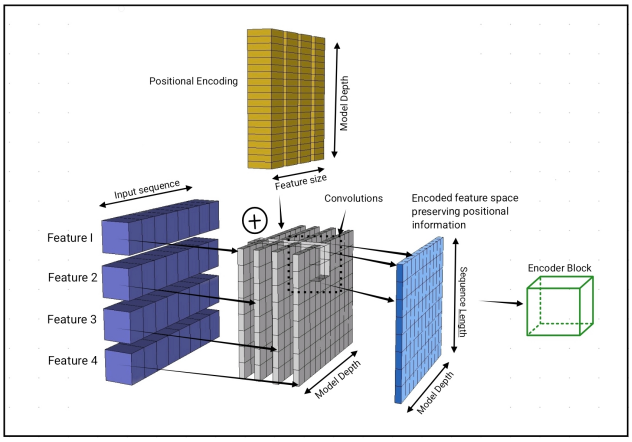

Figure 1: Proposed hybrid architecture encoding multiple features and corresponding positional information using Convolutional Layers.

3 Proposed Model for Long Sequence Time Series Forecasting

3.1 Challenges with Transformer Models in LSTF

Transformer models, recognized for their efficiency in processing sequences, have become the backbone for large-scale language modeling. However, their application to LSTF introduces significant challenges. One major limitation is their quadratic time complexity with respect to sequence length, which becomes a bottleneck when handling multivariate time sequences. The multivariate nature of time series data increases computational overhead, as each time step in the sequence contains multiple features that must be jointly modeled.

Additionally, the positional encoding mechanism used in transformers is inherently designed for univariate sequences. Positional encodings are added to the token embeddings in a way that preserves order information, but this approach is less effective for multivariate time series. In such cases, each time step consists of a vector of features, and the transformer architecture struggles to handle the multivariate relationships and long-range dependencies effectively. This architectural limitation

motivates the need for alternative methods that can handle the complexity of multivariate time series forecasting.

3.2 Hybrid CNN-Transformer Model for LSTF

To overcome these limitations, hybrid model was proposed that combines the strengths of Convolutional Neural Networks (CNNs) with transformer architectures. This approach transforms the multivariate time series data into a structured format that allows the transformer to effectively capture both short-term and long-term dependencies. Proposed model addresses both the challenges of computational complexity and the positional encoding mismatch in transformers.

3.3 Encoder Module

The encoder begins by receiving an input sequence of token indices, denoted as 1×120, representing the embedded data. Each token is mapped to a 512-dimensional embedding, creating an initial tensor of size $N \times 120 \times 512$, where N refers to the feature size. To preserve the temporal ordering of the sequence, positional encoding is added to the token embeddings. The positional encoding has the dimensions 512×1, ensuring that the model can distinguish between the positions of tokens within the sequence.

Once positional encodings are applied, the embedded sequence is passed through a series of convolutional layers. The convolutional layers utilize 10×10 CNN filters, allowing the model to extract local patterns from the input sequence across both the temporal and channel dimensions. The convolutional layers are essential in identifying high-level features in the sequence that can later be processed by the attention mechanism.

Following the convolutional layers, a global self-attention mechanism is applied to the feature map. Self-attention allows the model to capture dependencies across the entire sequence by weighing the importance of each token in relation to the others.

3.4 Decoder Module

The decoder takes as input a token-indexed sequence of size 1×120, which corresponds to the expected output sequence. Like the encoder, the tokens are embedded into a 512-dimensional space, resulting in an input tensor of size $N \times 120 \times 512$. A positional encoding of size 512×1 is added to the embeddings to maintain temporal information throughout the sequence. The decoder begins by applying causal attention, which ensures that at each time step, the model only attends to tokens from previous time steps. This ensures that the predictions made at time t are conditioned solely on the past, enforcing a causality constraint and allowing the model to generate coherent sequences. Following causal attention, a cross-attention mechanism is applied. Cross-attention allows the decoder to attend to the encoder's output feature map, which contains the processed information from the input sequence. This enables the decoder to make use of both past information and the information derived from the input sequence. The result is a more informed generation process that considers the entire context of the input sequence. As with the encoder, residual connections and normalization layers are utilized to stabilize training and enhance the flow of gradients through the model.

The final step in the decoder involves selecting the token with the highest probability at each time step, forming the output sequence. This process is iteratively applied across the sequence to generate the final prediction.

3.5 Summary of the Architecture

The overall architecture is a combination of convolutional layers for local feature extraction and attention mechanisms for capturing global dependencies. The encoder as shown in 2 transforms the input sequence into a feature map that captures both local and global features. The decoder uses this feature map, along with causal and cross-attention mechanisms, to generate the output sequence. The model is designed to handle long-range dependencies efficiently while maintaining computational efficiency through the use of CNNs and attention mechanisms.

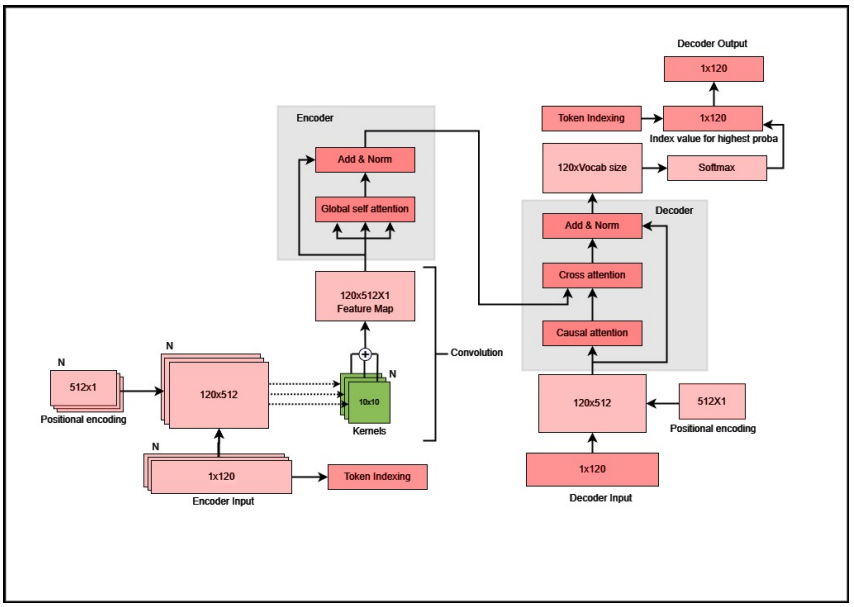

Figure 2: Overall architecture flow diagram.

4 Results

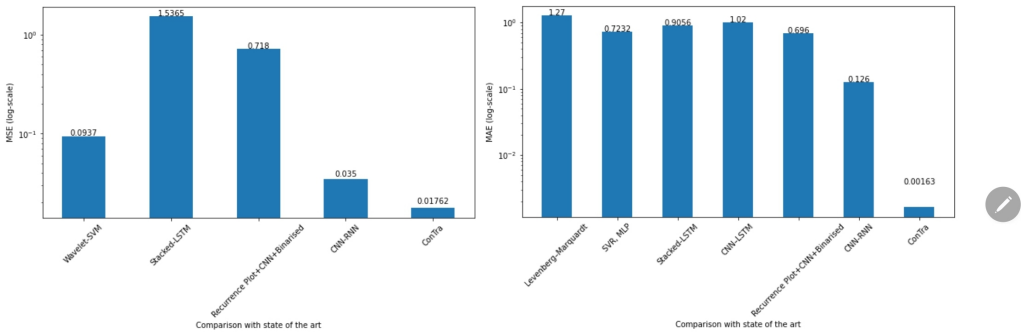

Figure 3: Results

The performance of the proposed architecture was rigorously evaluated using the Jena dataset for 24-hour forecasting. The model achieved the following metrics: Root Mean Square Error ($RMSE$) = 0.00799, Mean Absolute Error (MAE) = 0.00163, Mean Squared Error (MSE) = 0.01762, and a coefficient of determination (R^2) = 0.9991.

As illustrated in Figure 3, the proposed architecture demonstrates a significant improvement over existing methods on the Jena dataset across all metrics. These results reflect a high level of precision in the model's forecasts when compared against the test samples. The R^2 value of 0.9991 indicates that the model accounts for approximately 99.91% of the variability in the target variable, underscoring the model's robustness and excellent predictive capabilities.

5 Conclusion

This paper introduced a novel hybrid model combining Convolutional Neural Networks(CNNs) and Transformer architectures for long-sequence time series forecasting. The proposed approach effectively addresses the challenges of capturing both local and long-range dependencies in multivariate meteorological data, a common issue faced by traditional methods such as ARIMA, LSTM, and hy-

brid CNN-LSTM models. By leveraging CNNs for feature extraction and Transformers for capturing long-term dependencies, proposed model demonstrated superior performance over state-of-the-art models.

The results showed significant improvements in predictive accuracy, with the model achieving a Root Mean Squared Error (RMSE) of 0.00799, a Mean Absolute Error (MAE) of 0.00163, and a coefficient of determination (R^2) of 0.9991. These results underscore the effectiveness of the proposed hybrid CNN-Transformer model in handling complex, multivariate time series data, offering substantial improvements in both short-term and long-term weather forecasting tasks.

References

Jena Climate Dataset. Weather time series dataset recorded at the weather station of the max planck institute for biogeochemistry in jena, germany, 2022.

Francisco J Doblas-Reyes, Javier García-Serrano, Fabian Lienert, Aida Pintó Biescas, and Luis RL Rodrigues. Seasonal climate predictability and forecasting: status and prospects. *Wiley Interdisciplinary Reviews: Climate Change*, 4(4):245–268, 2013.

Ömer Altan Dombaycı and Mustafa Gölcü. Daily means ambient temperature prediction using artificial neural network method: A case study of turkey. *Renewable Energy*, 34(4):1158–1161, 2009.

Dusan Fister, Jorge Pérez-Aracil, César Peláez-Rodríguez, Javier Del Ser, and Sancho Salcedo-Sanz. Accurate long-term air temperature prediction with machine learning models and data reduction techniques. *Applied Soft Computing*, 136:110118, 2023.

Moinul Hossain, Banafsheh Rekabdar, Sushil J Louis, and Sergiu Dascalu. Forecasting the weather of nevada: A deep learning approach. In *2015 international joint conference on neural networks (IJCNN)*, pages 1–6. IEEE, 2015.

Jingwei Hou, Yanjuan Wang, Ji Zhou, and Qiong Tian. Prediction of hourly air temperature based on cnn–lstm. *Geomatics, Natural Hazards and Risk*, 13(1):1962–1986, 2022.

Ozgur Kisi and Jalal Shiri. Prediction of long-term monthly air temperature using geographical inputs. *International Journal of Climatology*, 34(1):179–186, 2014.

Chengsi Li, Mengyisong Zhao, Yilong Liu, and Fangzhou Xu. Air temperature forecasting using traditional and deep learning algorithms. In *2020 7th International conference on information science and control engineering (ICISCE)*, pages 189–194. IEEE, 2020.

Cong Li, Yaonan Zhang, and Guohui Zhao. Deep learning with long short-term memory networks for air temperature predictions. In *2019 International conference on artificial intelligence and advanced manufacturing (AIAM)*, pages 243–249. IEEE, 2019.

Xiaohong Liu, Shujuan Yuan, and Li Li. Prediction of temperature time series based on wavelet transform and support vector machine. *J. Comput.*, 7(8):1911–1918, 2012.

Sancho Salcedo-Sanz, RC Deo, Leopoldo Carro-Calvo, and Beatriz Saavedra-Moreno. Monthly prediction of air temperature in australia and new zealand with machine learning algorithms. *Theoretical and applied climatology*, 125:13–25, 2016.

A Utku and U Can. An efficient hybrid weather prediction model based on deep learning. *International Journal of Environmental Science and Technology*, 20(10):11107–11120, 2023.