International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

MS-CLSTM: A Novel Deep Learning Approach for Forecasting Atmospheric Temperature Indices at Intervals

Surabhi Anuradha[1], Surabhi Sivakumar[*2], Pothabathula Naga Jyothi[3]

Submitted: 29/01/2024 **Revised:** 07/03/2024 **Accepted:** 15/03/2024

Abstract: Time series forecasting plays a vital role in advancing fields like finance, economics, meteorology, and stock market analysis. It enables the prediction of future values by examining patterns in historical data. Forecasting becomes a great deal with recent strides in accuracy propelled by various deep learning techniques. In this study, we introduced a multi-stacked ConvLSTM (MS-CLSTM) model designed for precise periodic forecasting of atmospheric temperature indices. The model's performance was evaluated against Long Short-Term Memory (LSTM) and Temporal Convolution Network (TCN) models across diverse historical input windows and target prediction scenarios. Assessment metrics such as Mean Square Error (MSE) and Mean Absolute Error (MAE) were employed to gauge accuracy, particularly in predicting twelve-hour periodic temperature projections using a three-day historical temperature window as input. Our findings revealed a substantial improvement in performance during validation, showcasing a 42% reduction in MSE and a 20% decrease in MAE compared to LSTM. Additionally, when compared to TCN, our proposed model exhibited a 12% decrease in MSE and a 5% drop in MAE. Notably, the model consistently demonstrated strong performance across various input window sizes, encompassing historical information ranging from two to five days, and in predicting varying target scenarios.

Keywords: *forecasting, time series, temporal convolution, structured encoding, multi-stacking*

1. Introduction

In the vast landscape of data science, where insights are sought amidst the ebb and flow of information, one powerful technique stands out – Time Series Forecasting. In striving to make sense of sequential data, predicting future trends, and harnessing the predictive potential hidden in temporal patterns, time series forecasting emerges as a guiding light [1].

Time series forecasting plays a pivotal role across diverse industries. Beyond energy and finance, its applications extend to optimizing retail inventory, enhancing traffic management for urban planning, aiding healthcare resource allocation, predicting weather patterns for disaster preparedness, optimizing manufacturing and supply chains, facilitating human resources and workforce planning, managing energy consumption in smart buildings, forecasting demand in e-commerce, predicting agricultural yields for farmers, and aiding telecommunications companies in network planning. Additionally, time series forecasting enables electricity companies to formulate effective energy policies by anticipating future consumption trends, contributing to efficient resource allocation. Similarly, corporations leverage accurate predictions of future stock prices to mitigate investment risks and make informed financial decisions, demonstrating the broad-reaching impact of time series forecasting in guiding strategic planning and operational excellence across various sectors [2], [3], [4], [5], [6], [7], [8], [9], [10]. Despite being a subject of study for decades, time series forecasting remains a challenging and active research problem, primarily due to the inherent complexity of time series data. Traditional methods, such as autoregressive models (AR), moving average models (MA), and autoregressive integrated moving average models (ARIMA) [11], approach forecasting by assuming linear relationships with observed values. However, this simplification often falls short in addressing the intricacies of real-world time series data.

The advent of deep learning techniques has significantly impacted time series forecasting. Deep neural networks (DNN), in particular, have become widely adopted for their ability to capture non-linear relationships and temporal dependencies inherent in time series data [12]. Among DNN architectures, convolutional neural networks (CNN) excel at identifying local patterns in short time series subsequence's, such as seasonality and trend, while recurrent neural networks (RNN) specialize in capturing long-term or mid-term temporal dynamics of the entire time series [13], [14], [15].

In practice, the combination of both CNN and RNN has proven effective in capturing diverse temporal information for more accurate forecasting results. This hybrid approach addresses the limitations of solely relying on one architecture. For example, researchers in genomics have successfully employed a hybrid neural network by stacking CNN with RNN for DNA sequence prediction [16]. CNN identifies short and recurring sequence motifs representing biological function units, while RNN, specifically long short-term memory (LSTM), learns the spatial arrangement of these motifs. Elaborating on this approach, the hybrid model ensures a comprehensive understanding of both local and long-term temporal patterns, enhancing its predictive

1Dept. of CSE(AIML), KMIT, Telangana, INDIA
ORCID ID : 0009-0008-6091-859X
2 Dept of Chemistry, ANITS, Andhra Pradesh, INDIA
ORCID ID : 0000-0002-0173-1169
3 Dept. of CSE, GITAM School of Technology, Andhra Pradesh, INDIA
ORCID ID : 0000-0002-6397-5951
** Corresponding Author Email: sivakumar.chemistry@anits.edu.in*

capabilities. This integration of multiple neural network architectures showcases the versatility and adaptability of deep learning in tackling the complexities of time series forecasting, further emphasizing its potential in diverse fields of study and application. The ConvLSTM model combines the benefits of CNN and LSTM, enabling the learning of long-term dependencies and extraction of time-invariant features, stated by Shi et al. [17].

Current Study endeavours to formulate an innovative model suggesting a sophisticated neural network architecture featuring multiple layers for systematic encoding and structured information learning from input data. The benchmark model utilized is a complex configuration of Convolutional Long Short-Term Memory (ConvLSTM) with multiple stacked layers (MS-CLSTM). The model's effectiveness is evaluated through experiments employing weather time series dataset. To thoroughly scrutinize the efficacy of our developed model, we conducted comparisons with leading models from existing literature tailored to each specific time series. Our results consistently demonstrate the superiority of the proposed methodology, showcasing enhanced forecast accuracy in contrast to baseline methods such as LSTM and TCN.

2. Experimental Setup

2.1 Dataset

Weather time series dataset recorded at Weather Station, Max Planck Institute for Biogeochemistry in Jena, Germany. The Time frame considered was 01-01-2016 to 01-11-2023. The dataset consists of 411592 rows across 22 different features of Jena climate data such as 'Date Time', 'p (mbar)', 'T (degC)', 'Tpot (K)', 'rh (%)', and so on. In this study, our primary focus was on analyzing temperature patterns. We specifically examined the historical data, aiming to predict future temperatures. Consequently, our attention was directed towards only two key attributes: 'Date Time' and 'T (degC)'.

2.2 Input-label Window Size

Defining the input-label window size involves configuring historical input data and its corresponding output sequence in a time series forecasting model, particularly for a univariate multi-step time prediction. In this context, the term "univariate" signifies the model's focus on predicting a single variable or time series, while "multi-step" denotes the prediction of multiple future time steps. For instance, in a scenario of univariate multi-step time prediction with a window size of 'n', the model utilizes 'n' consecutive past values of a single variable as input to predict 'm' future values. The input sequence has a length of 'n', and the output sequence comprises the predicted values for the next 'm' time steps.

In the weather dataset, observations are recorded at 10-minute intervals, yielding six observations per hour and 144 observations per day. To forecast the temperature for the next 12 hours, a predictive strategy is employed. This strategy involves using a window of the last 720 observations (equivalent to the number of days, considering the 144 observations per day) as input data. The corresponding 72 observations following this window are used as labels for training the predictive model. This approach enables the model to learn patterns and relationships from the historical temperature data, allowing it to make predictions for the temperature in the upcoming 12-hour period based on the past 5 days of information.

3. Methods Applied

3.1 Long Short-Term Memory (LSTM)

LSTM, a specialized recurrent neural network structure, has demonstrated stability and efficacy in modeling long-range dependencies across various studies [18], [19] when applied to general-purpose sequence modeling. A key innovation of LSTM lies in its incorporation of a memory cell, denoted as c_t, which functions as an accumulator of state information. This memory cell is accessed, written to, and cleared through self-parameterized controlling gates. Upon receiving a new input, the information is accumulated in the cell if the input gate it is activated. Additionally, the past cell status c_{t-1} may undergo "forgetting" if the forget gate f_t is active. The determination of whether the latest cell output c_t is propagated to the final state h_t is further regulated by the output gate o_t. With the symbol '$\odot$' representing the element-wise product, the complete LSTM architecture is defined as

$$i_t = \sigma\,(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \odot c_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma\,(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \odot c_{t-1} + b_f) \tag{2}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\,(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma\,(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \odot c_t + b_o) \tag{4}$$

$$h_t = o_t \odot \tanh(c_t) \tag{5}$$

3.2 Temporal Convolution Network (TCN)

The domain of sequence modeling within the realm of deep learning has traditionally been dominated by recurrent neural network architectures like LSTM and Gated Recurring Unit (GRU). S. Bai et al. argued that this conventional perspective is outdated, proposing that convolutional networks should be given serious consideration as primary candidates for modeling sequential data [20]. They demonstrated that convolutional networks can outperform RNNs in numerous tasks, overcoming typical challenges associated with recurrent models, such as the exploding/vanishing gradient problem and limited memory retention. Additionally, opting for a convolutional network over a recurrent one can yield performance enhancements by enabling parallel computation of outputs [21].

Lea et al. stated that, TCN possesses unique features, including causal convolutions, ensuring no information leakage from the future to the past. Additionally, TCN exhibits the ability to effectively analyze extensive historical data for future predictions. This is achieved through a combination of deep networks, complemented by residual layers and dilated convolutions, allowing for an in-depth examination of past data to inform future forecasts [22]. The TCN adopts a 1D fully-convolutional network (FCN) architecture, as introduced by Long et al. [23]. In this architecture, each hidden layer maintains the same length as the input layer, and zero padding of length (kernel size − 1) is incorporated to ensure successive layers retain the same length as their predecessors. The fulfillment of this objective relies on the utilization of causal convolutions, where the convolution operation at time 't' involves only elements from time 't' and earlier in the preceding layer.

A fundamental constraint of basic causal convolution lies in its ability to consider a history size linearly dependent on the network's depth. However, the receptive field sizes achievable with causal convolution remain limited unless a substantial number of layers are stacked. This limitation becomes particularly challenging in sequence tasks, given the computational burden associated with causal convolution. To overcome these challenges, dilated convolutions, as proposed by Yu and Koltun [24], offer the advantage of creating an exponentially large receptive field without the necessity of an excessive layer count. In a more formal context, considering a 1-D sequence input x ∈ Rn and a filter f: {0, ..., k − 1} → R, the dilated convolution operation F on the sequence's element s is defined as

$$F(s) = (x *_d \ f)(s) = \sum_{i=0}^{k-1} f(i).\, x_{s-d.i} \tag{6}$$

Here, *d* represents the dilation factor, *k* denotes the filter size, and $s - d \cdot i$ incorporates the past direction into the equation.

To achieve a sufficiently large receptive field size in TCN, it is essential to increase both network depth *n* and the filter size *k* along with the dilation factor *d*. Empirical observations indicate that an effective architecture involves making the network deep and narrow, implying the stacking of numerous layers and selecting a slim filter size. Additionally, the integration of residual connections, as demonstrated in residual networks [25], has shown significant effectiveness in training deep networks by utilizing skip connections throughout the architecture. A residual block, as introduced by He et al. [26], comprises a branch that extends to a sequence of transformations F, and the results are summed with the input *x* of the block is defined as

$$o = Activation\ (x + F(x)) \tag{7}$$

3.3 Proposed model: MS-CLSTM

ConvLSTM, a variant of recurrent neural networks, is designed for spatio-temporal prediction with convolutional structures integrated into both input-to-state and state-to-state transitions. In the context of ConvLSTM, the prediction of the future state of a specific cell within a grid is influenced by the inputs and previous states of its neighboring cells. This is accomplished through the utilization of convolutional operators in both the state-to-state and input-to-state transitions [17].

The primary limitation of LSTM is failing to encode spatial information. To overcome this challenge, a distinctive aspect of our design involves representing all inputs (X1, ..., Xt), cell outputs (C1, ..., Ct), hidden states (H1, ..., Ht), and gates (it, ft, ot) of the ConvLSTM as 3D tensors, with the last two dimensions corresponding to spatial dimensions (rows and columns). To provide a clearer perspective on the inputs and states, one can envision them as vectors positioned on a spatial grid. In this configuration, the ConvLSTM determines the future state of a specific cell within the grid based on the inputs and past states of its local neighbors. This is efficiently accomplished by incorporating a convolution operator in both the state-to-state and input-to-state transitions. The ConvLSTM architecture, where '∗' denotes the convolution operator and '⊙' denotes the element-wise product is defined as

$$i_t = \sigma\ (W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci}\ \odot\ C_{t-1}\ + b_i) \tag{8}$$

$$f_t = \sigma\ (W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf}\ \odot\ C_{t-1}\ + b_f) \tag{9}$$

$$C_t =\ f_t\ \odot C_{t-1} +\ i_t\ \odot \tanh\ (W_{xc} * X_t + W_{hc} * H_{t-1} +\ b_c\ ) \tag{10}$$

$$o_t = \sigma\ (W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co}\ \odot\ c_t\ + b_o) \tag{11}$$

$$h_t =\ o_t\ \odot \tanh(c_t) \tag{12}$$

The proposed model MS-CLSTM, as illustrated in Fig. 1, incorporates a sequence of four stacked ConvLSTM layers. The ConvLSTM2D layer in this model combines convolutional operations with the long short-term memory (LSTM) architecture, making it well-suited for handling spatiotemporal data, such as sequences of images. The initial ConvLSTM layer applies a convolutional LSTM operation with 64 filters and a kernel size of (10, 1). Using 'same' padding ensures that the output maintains the same spatial dimensions as the input. By setting 'return_sequences' to True, this layer provides the complete sequence of outputs for each input sequence. Dropout is incorporated to prevent overfitting, while batch normalization aids in stabilizing and expediting the training process.

The second ConvLSTM layer is akin to the first, but with return_sequences set to False, indicating that it only yields the output for the final timestep of the input sequence. This can be advantageous for dimensionality reduction. The third and fourth ConvLSTM layers mirror the first one, returning the full sequence of outputs. The TimeDistributed Dense layer applies a dense operation with ReLU activation to each timestep independently. This is commonly employed to process the output sequence of preceding layers autonomously. The final dense layer, utilizing linear activation, generates the ultimate output. To summarize, the ConvLSTM2D layers analyze the spatiotemporal features in the input sequences, capturing both spatial and temporal dependencies. The amalgamation of convolutional and LSTM operations empowers the model to comprehend intricate patterns in the input data, and subsequent dense layers map these features to produce the final output. The inclusion of dropout and batch normalization serves to regularize and stabilize the training process.

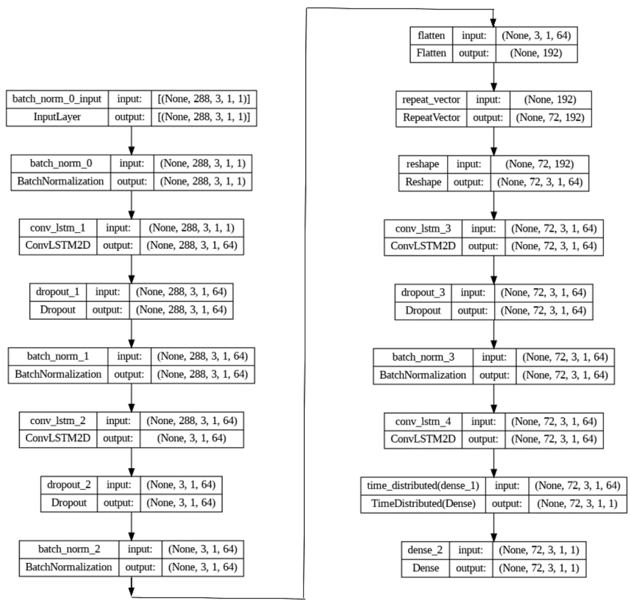

Fig. 1. Proposed MS-CLSTM Model Architecture.

4. Performance Measures

To assess the effectiveness of our forecasting methods, we employ two distinct Goodness-of-Fit (GoF) measures. The initial measure is the Mean Square Error (MSE) is characterized as

$$MSE = \frac{1}{T}\sum_{t=1}^{T}\left(Y_t - \hat{Y}_t\right)^2 \tag{13}$$

Given a forecast $\hat{Y}_t$ for a temperature Y_t. we calculate the Mean Absolute Error (MAE), which assesses a method's average absolute error relative to the average absolute error of the seasonal naive forecast. The MAE is defined as

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\left(Y_t - \hat{Y}_t\right) \tag{14}$$

5. Results and Discussion

To comprehensively assess the behavior of our models, we initiated the analysis by employing the LSTM model on our dataset. Our exploration entailed executing the model with different window sizes, spanning from 2 days to 5 days of historical data, predicting the immediate 12 hours into the future, and also forecasting outcomes after a 2-day interval. Subsequently, we conducted similar experiments with both TCN and ConvLSTM models, maintaining the same window sizes. Throughout each model iteration, performance is evaluated by measuring key metrics, MSE and MAE. Table 1 provides a comprehensive overview of MSE and MAE values for three different models LSTM, TCN, and MS-CLSTM across multiple training epochs and past history lengths.

Utilizing a 2-day past history as input, the MS-CLSTM model consistently enhances its predictive capabilities, showcasing a gradual decline in training MSE from 0.253 in Epoch 1 to 0.09 in Epoch 4. Simultaneously, the model demonstrates proficient generalization, evident in the reduction of validation MSE from 0.118 to 0.074 across the same epochs, underscoring its adept learning and forecasting prowess. Compared to LSTM, MS-CLSTM achieved a significant 36.21% reduction in validation loss, signifying superior performance, and demonstrated a notable 9.76% reduction when compared to TCN, highlighting its effectiveness in the forecasting task.

Additionally, the MAE loss values for the models reveal distinct patterns. The LSTM model exhibited a gradual decrease in both training and validation MAE, reaching 0.276 and 0.265, respectively. In contrast, the TCN model demonstrated a reduction in both training and validation MAE to 0.231 and 0.216, while the MS-CLSTM model displayed the lowest MAE values at 0.227 for training and 0.202 for validation. These results suggest that both TCN and MS-CLSTM models outperformed the LSTM model in minimizing absolute errors, with the MS-CLSTM model showcasing the most effective learning and generalization capabilities, achieving the lowest MAE values for both training and validation.

At the culmination of Epoch 4 with a 3-day past history and targeting the next 12 hours, a comparative analysis of MSE and MAE losses in the validation set reveals distinctive performances among LSTM, TCN, and MS-CLSTM models. LSTM demonstrates effective learning with a reduction in validation MSE from 0.279 in Epoch 1 to 0.125 in Epoch 4 and a corresponding decrease in MAE from 0.396 to 0.256. TCN exhibits improved performance, reflected in the reduction of validation MSE from 0.318 to 0.083 and a decrease in MAE from 0.418 to 0.214. Notably, the MS-CLSTM model surpasses both counterparts, achieving the lowest validation MSE of 0.073 in Epoch 4, signifying superior accuracy. The MS-CLSTM model also minimizes absolute errors effectively, with the lowest MAE of 0.204. MS-CLSTM exhibited a 20.31% reduction in validation MAE and 41.60%. reduction in MSE compared to LSTM, emphasizing its proficiency in minimizing absolute errors. In comparison to TCN, MS-CLSTM demonstrated a 12.05% reduction in validation MSE and a 4.67% reduction in validation MAE, underscoring its effectiveness in both accuracy and precision for the specific time series forecasting task. These results highlight the advantageous capabilities of the MS-CLSTM model in achieving superior validation performance compared to both LSTM and TCN.

Fig. 2 illustrates the training and validation Mean Squared Error (MSE) loss for the models under two different past history lengths: (a) 2 days and (b) 3 days. The visual representation provides a comparative view of how each model's MSE loss evolves during training and testing across multiple epochs for both past history scenarios. Fig. 3 to 5 display sample predictions generated by each of the models.

Table 1. Model Performance Metrics for 2 Days and 3 Days Past History for next twelve hours prediction

			MSE				***MAE***			
Past History	***Model***	***Loss***	***Epoch 1***	***Epoch 2***	***Epoch 3***	***Epoch 4***	***Epoch 1***	***Epoch 2***	***Epoch 3***	***Epoch 4***
2 days (288 observations)	LSTM	Training	0.453	0.187	0.163	0.126	0.506	0.339	0.316	0.276
		Validation	0.261	0.211	0.157	0.116	0.394	0.355	0.308	0.265
	TCN	Training	0.324	0.155	0.116	0.1	0.837	0.299	0.258	0.231
		Validation	0.271	0.115	0.089	0.082	0.389	0.257	0.224	0.216
	MS-CLSTM	Training	0.253	0.125	0.096	0.09	0.486	0.27	0.232	0.227
		Validation	0.118	0.085	0.075	0.074	0.257	0.217	0.208	0.202
3 days (432 observations)	LSTM	Training	0.401	0.17	0.127	0.096	0.479	0.321	0.275	0.239
		Validation	0.279	0.228	0.158	0.125	0.396	0.342	0.29	0.256
	TCN	Training	1.586	0.162	0.115	0.101	0.784	0.304	0.256	0.239
		Validation	0.318	0.121	0.095	0.083	0.418	0.259	229	0.214
	MS-CLSTM	Training	0.721	0.128	0.095	0.087	0.422	0.275	0.233	0.223
		Validation	0.094	0.092	0.08	0.073	0.232	0.225	0.213	0.204

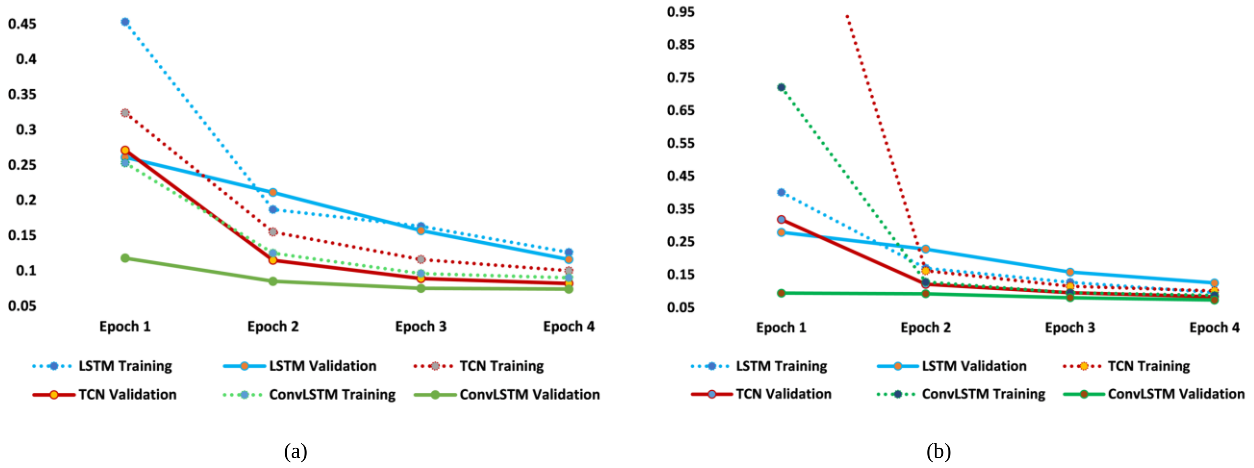

Fig. 2. Training and Validation MSE Loss for the models with (a) two day historical data; (b) three day historical data.

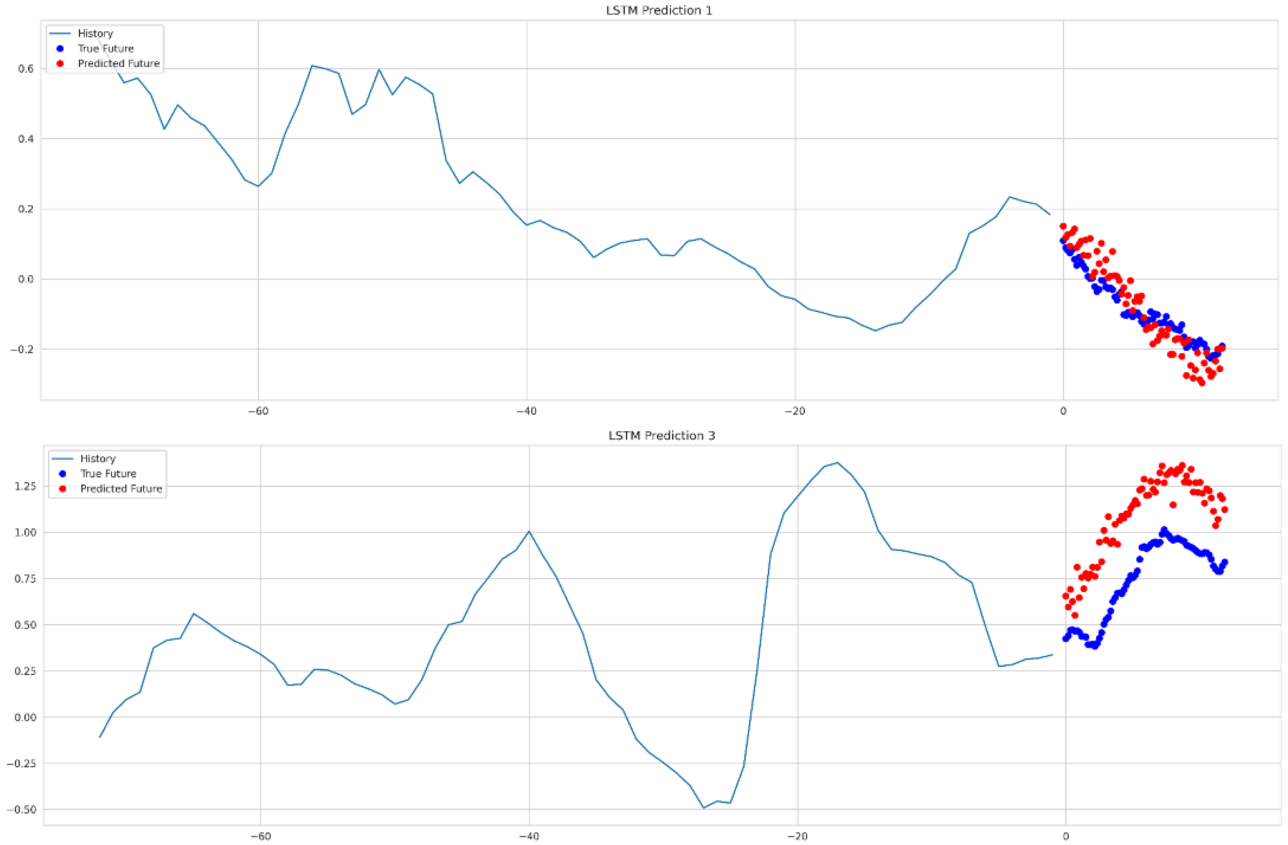

Fig. 3. Sample predictions using LSTM model with a three-day historical input.

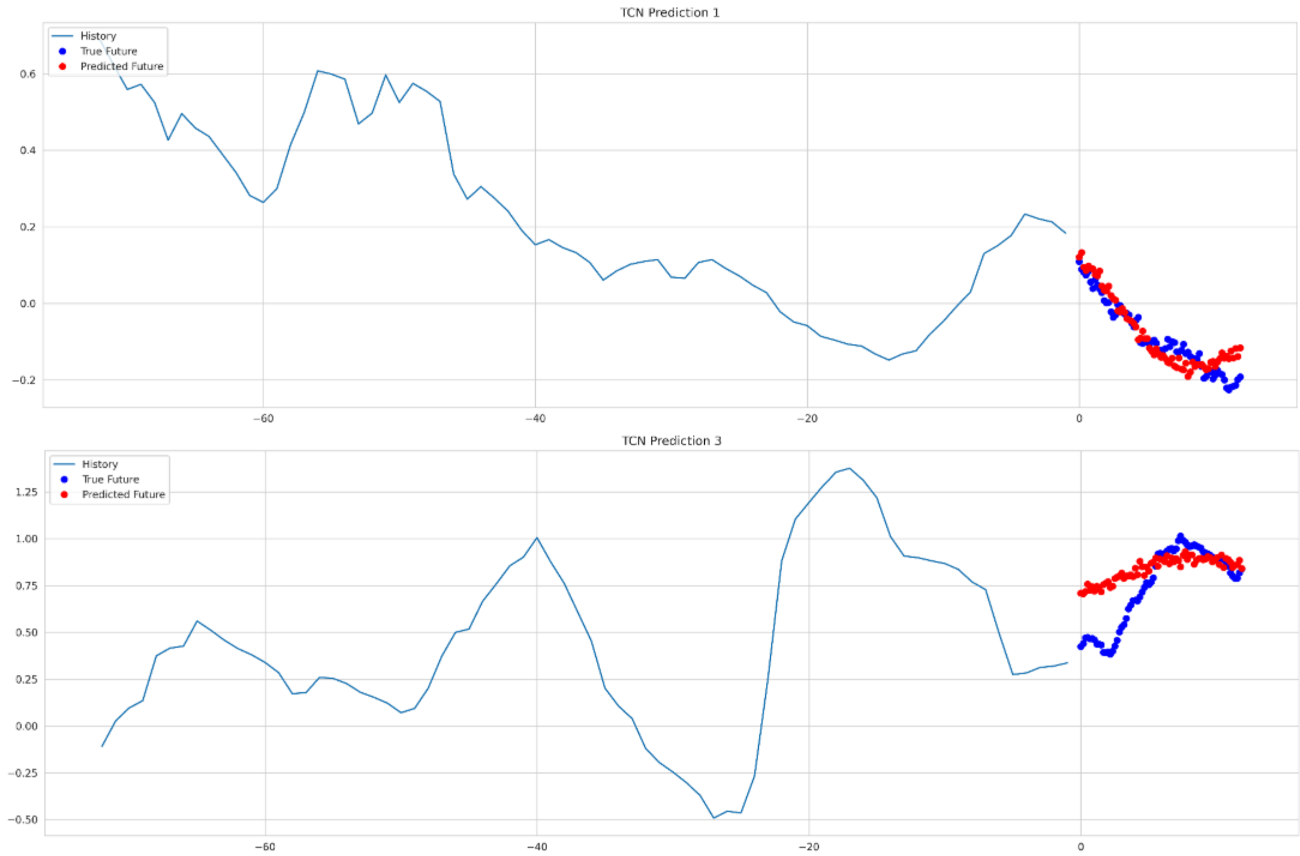

Fig. 4. Sample predictions using TCN model with a three-day historical input.

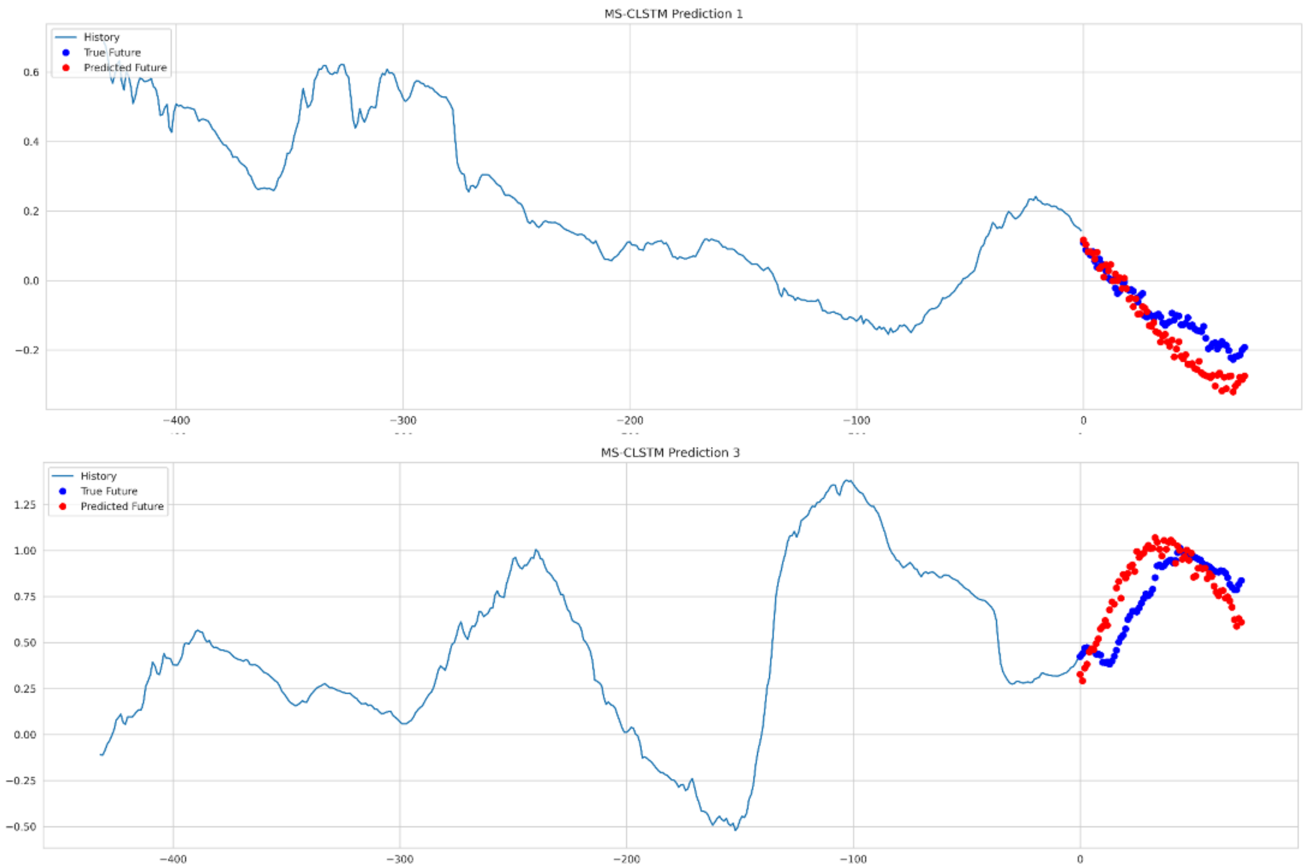

Fig. 5. Sample predictions using MS-CLSTM model with a three-day historical input.

6. Conclusion

In the conducted study, a versatile model was proposed by incorporating multiple stacked ConvLSTM layers named MS-CLSTM, establishing a structured encoding-forecasting architecture. This model not only successfully addressed the challenge of nowcasting but also showcased its adaptability to a broader range of spatiotemporal sequence forecasting problems. MS-CLSTM, presents a robust foundation for tackling intricate challenges in the realms of temporal and spatial forecasting. This research opens avenues for further exploration and application of these methodologies across diverse domains, promising advancements in predictive modeling for future events, complex spatiotemporal consequences of climate changes.

Author contributions

Anuradha Surabhi: Conceptualization, Methodology, Coding, Draft preparation, **Sivakumar Surabhi:** Data curation, Code and Result validation, reviewing modifications, **NagaJyothi Pothabathula:** Investigation, Writing, Reviewing and Editing.

Conflict of Interest

The authors declare No conflict of interest.

References

[1] Zhou, Haoyi, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," In Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 12, pp. 11106-11115. 2021.

[2] Wang, Kun, Chenhan Xu, Yan Zhang, Song Guo, and Albert Y. Zomaya. "Robust big data analytics for electricity price forecasting in the smart grid." IEEE Transactions on Big Data 5, no. 1 (2017): 34-45.

[3] Li, Audeliano Wolian, and Guilherme Sousa Bastos. "Stock market forecasting using deep learning and technical analysis: a systematic review." IEEE access 8 (2020): 185232-185242.

[4] Lefkowitz, Brett, and Moran Bodas. "Threat perception of climate change & public preparedness for extreme weather-related events in Bermuda-A cross-sectional study." International journal of disaster risk reduction 97 (2023): 104030.

[5] Bilal, Muhammad, Hyeok Kim, Muhammad Fayaz, and Pravin Pawar. "Comparative analysis of time series forecasting approaches for household electricity consumption prediction." arXiv preprint arXiv:2207.01019 (2022).

[6] Sufian, Md Abu, and Jayasree Varadarajan. "Enhancing Prediction and Analysis of UK Road Traffic Accident Severity Using AI: Integration of Machine Learning, Econometric Techniques, and Time Series Forecasting in Public Health Research." arXiv preprint arXiv:2309.13483 (2023).

[7] Mandrikova, Oksana, Nadezhda Fetisova, and Yuriy Polozov. "Hybrid model for time series of complex structure with ARIMA components." Mathematics 9, no. 10 (2021): 1122.

[8] Al-Qazzaz, Redha Ali, and Suhad A. Yousif. "High performance time series models using auto autoregressive integrated moving average." Indones. J. Electr. Eng. Comput. Sci 27 (2022): 422-430.

[9] Sethi, Jasleen Kaur, and Mamta Mittal. "Analysis of air quality using univariate and multivariate time series models." In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 823-827. IEEE, 2020.

[10] Sagheer, Alaa, and Mostafa Kotb. "Time series forecasting of petroleum production using deep LSTM recurrent networks." Neurocomputing 323 (2019): 203-213.

[11] Junior, Paulo Rotela, Fernando Luiz Riêra Salomon, and Edson de Oliveira Pamplona. "ARIMA: An applied time series forecasting model for the Bovespa stock index." Applied Mathematics 5, no. 21 (2014): 3383.

[12] Lim, Bryan, and Stefan Zohren. "Time-series forecasting with deep learning: a survey." Philosophical Transactions of the Royal Society A 379, no. 2194 (2021): 20200209.

[13] Akşehir, Zinnet Duygu, and Erdal Kiliç. "How to handle data imbalance and feature selection problems in CNN-based stock price forecasting." IEEE Access 10 (2022): 31297-31305.

[14] Liu, Shuai, Hong Ji, and Morgan C. Wang. "Nonpooling convolutional neural network forecasting for seasonal time series with trends." IEEE transactions on neural networks and learning systems 31, no. 8 (2019): 2879-2888.

[15] Bi, Jing, Xiang Zhang, Haitao Yuan, Jia Zhang, and MengChu Zhou. "A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM." IEEE Transactions on Automation Science and Engineering 19, no. 3 (2021): 1869-1879.

[16] Quang, Daniel, and Xiaohui Xie. "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences." Nucleic acids research 44, no. 11 (2016): e107-e107.

[17] Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." Advances in neural information processing systems 28 (2015).

[18] Graves, Alex. "Generating sequences with recurrent neural networks." arXiv preprint arXiv:1308.0850 (2013).

[19] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.

[20] Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv preprint arXiv:1803.01271 (2018).

[21] Chen, Jungan, Danjiang Chen, and Gaoping Liu. "Using temporal convolution network for remaining useful lifetime prediction." Engineering Reports 3, no. 3 (2021): e12305.

[22] Lea, Colin, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. "Temporal convolutional networks for action segmentation and detection." In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 156-165. 2017.

[23] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440. 2015.

[24] Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." arXiv preprint arXiv:1511.07122 (2015).

[25] Duc, Tho Nguyen, Chanh Tran Minh, Tan Phan Xuan, and Eiji Kamioka. "Convolutional neural networks for continuous QoE prediction in video streaming services." IEEE Access 8 (2020): 116268-116278.

[26] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.