IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 July 2022, accepted 27 July 2022, date of publication 4 August 2022, date of current version 10 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3196381

RESEARCH ARTICLE

Short-Term Weather Forecasting Using Spatial Feature Attention Based LSTM Model

MASOOMA ALI RAZA SULEMAN[1] **AND S. SHRIDEVI**[2]
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[2]Centre for Advanced Data Science, Vellore Institute of Technology, Chennai 600127, India

Corresponding author: S. Shridevi (shridevi.s@vit.ac.in)

This work was supported by the Vellore Institute of Technology, India.

ABSTRACT Weather prediction and meteorological analysis contribute significantly towards sustainable development to reduce the damage from extreme events which could otherwise set-back the progress in development by years. The change in surface temperature is one of the important indicators in detecting climate change. In this research, we propose a novel deep learning model named Spatial Feature Attention Long Short-Term Memory (SFA-LSTM) model to capture accurate spatial and temporal relations of multiple meteorological features to forecast temperature. Significant spatial feature and temporal interpretations of historical data aligned directly to output feature helps the model to forecast data accurately. The spatial feature attention captures mutual influence of input features on the target feature. The model is built using encoder-decoder architecture, where the temporal dependencies in data are learnt using LSTM layers in the encoder phase and spatial feature relations in the decoder phase. SFA-LSTM forecasts temperature by simultaneously learning most important time steps and weather variables. When compared with baseline models, SFA-LSTM maintains the state-of-the-art prediction accuracy while offering the benefit of appropriate spatial feature interpretability. The learned spatial feature attention weights are validated from magnitude of correlation with target feature obtained from the dataset.

INDEX TERMS Sustainable environmental development, weather forecasting, recurrent neural network (RNN), long short-term memory (LSTM), spatial feature.

I. INTRODUCTION

Artificial Intelligence plays an important role in not only achieving sustainable development goals with respect to economy and society but also in achieving sustainable environmental goals by protecting and preserving biodiversity, in climate change, predicting extreme climatic conditions [1], evaluating ocean health [2], weather forecasting [3], [4], [5] and preventing spread of diseases [6], [7]. Now more than ever, environmental sustainability is becoming extremely crucial. The provisional World Meteorological Organization (WMO) State of the Global Climate 2021 report draws from the recent evidences to show how our earth is changing before our eyes. Weather prediction and meteorological analysis contribute significantly towards sustainable development to reduce the damage from extreme weather events, to decrease weather-related losses including protection of habitat, livelihood, economy which could otherwise set-back the progress in development by years.

Weather forecasting is the prediction of weather conditions for a given location and time through application of science, technology and principles of physics. The meteorological features such as atmospheric pressure, temperature, humidity, wind speed, precipitation of a given location collected over a time frame provides quantitative data describing the state of atmosphere at that particular point of time which is used for understanding the science of atmospheric processes to forecast future atmospheric state. Weather forecasting helps to plan the outcomes and influence of future weather conditions in our day-to-day activities. The ability to detect impending snow, rain, heat waves and floods help the public and government to plan and prevent its dreadful consequences.

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

The information about future weather conditions helps to maintain commercial, economic, environmental and social interests. For example, weather forecasts in agriculture helps the farmers to plan their harvests and work load, utility companies to purchase sufficient supplies of power and natural gas, inventories and stores to match the demand and supply of resources, public to plan their outdoor activities and government to communicate the weather warnings to general public to protect their life and property within sufficient time.

In the recent years, the number of climate monitoring systems has increased providing large amounts of hourly, daily, weekly, monthly and yearly weather-related information, and data remains transparent. This data is stored and provided so that other departments can utilize it by efficiently analyzing weather forecasts. The research is aimed at developing a machine learning platform for predictive modeling in the case of sustainable environmental management.

The proposed work targets to accelerate the discovery of new knowledge and optimize decision-making in sustainable environmental management. For that purpose, it is proposed to design and implement a machine learning (ML) pipeline that incorporates the necessary modules for a data-driven, accurate and effective weather forecasting. For effective forecasting, it is necessary to identify the interactions between meteorological features that indirectly contribute to climate change. An emphasis is made on temperature forecasting and building a deep neural network model to forecast weather while simultaneously learning interactions of different predictor variables. Therefore, in this paper, we propose a model for successful weather forecasting by considering the mutual influence of various meteorological features with target weather feature to be forecasted.

The major research contributions of our work are as follows:

- The proposed SFA-LSTM model is novel for multiple-input-single-output predictions in context of spatial feature interpretability in time series prediction. To the best of our knowledge, this is the first work on spatial feature time series prediction model where the spatial feature attention weight is aligned directly to the output feature.
- The model is trained to capture temporal patterns across multiple time steps and spatial feature interactions across multiple predictors to forecast the target variable. The target feature learns from both temporal and spatial feature contributions.
- Spatial feature attention mechanism is considered to grasp the quantitative mutual influence of input features on target feature.
- The proposed model will provide meaningful spatial feature interpretations which will be verified using domain knowledge

II. RELATED WORK

The weather is a dynamic, continuous, multi-dimensional and chaotic process [8]. Numerous methods have been developed to predict the weather. This section focuses on the work that has been done in the field of weather forecasting using machine learning and deep learning techniques and a special interest is taken on temperature forecast. Many researchers have tried to solve weather forecasting problem using different machine learning techniques [4], [5], [9], [10], [11] with successful results. Holmstrom *et al.* [9] proposed linear regression and functional regression which forecasts weather by searching historical weather patterns which are most similar to current weather pattern and Rasel *et al.* [5] performed a comparative study between Support Vector Regression [12] and Artificial Neural Networks [13] for temperature and rainfall prediction. The studies on deep learning neural networks [14], [15], deep belief networks [16], [17], [18] provide promising results with its ''deep'' architecture and higher learning ability in comparison to ''shallow'' machine learning models [14].

In the last decade, Recurrent Neural Networks (RNNs) have gained widespread attention and developed rapidly due to their powerful and effective modeling capabilities [19]. However, traditional RNN suffers from short term memory and vanishing gradient problems [20], [21], [22] which makes it difficult to capture long term dependencies, an important factor to capture historical relevant data over long time series to accurately predict the future weather. In the world of RNN, the Long-Short Term Memory (LSTM) based RNN overcomes the drawbacks of traditional RNN and formulates long-term dependencies between training samples [2], [23], [24], [25]. Shi *et al.* [26] proposed convLSTM network for precipitation nowcasting which consist convolutional structures in both input-to-state and state-to-state transitions which captures spatiotemporal relationships better than a fully connected LSTM network. A lightweight temporal convolutional neural network (TCN) has been developed [27] for short-to-medium range weather forecasting which is limited to regional forecasting and two weather parameters.

Karevan [24] proposed transductive LSTM (T-LSTM), a localized version of LSTM where the samples in the vicinity of test point have a higher impact on model fitting which is computationally expensive and not suitable for multivariate time series prediction. The drawback of transductive learning is the number of models that needs to be trained since the parameters of the model depend on individual test points. Kreuzer [28] proposed a new convLSTM model for local temperature forecasting where it uses six convolutional layers connected to an LSTM layer and a dense layer. Multi-stacked sequence to sequence LSTM model [29] to forecast temperature, wind speed and relative humidity and the proposed model could forecast weather with high accuracy. A similar approach was taken by Park [30] to restore the missed temperature data using four layered LSTM model which outperformed the deep neural network (DNN). Three DNNs, (Multi-Layer Perceptron) MLP, LSTM and CNN+LSTM were used by Roy [31] to forecast air temperature of weather station and the result indicated that prediction accuracy increases with increase in model complexity.

Several other models have been proposed based on LSTM-RNN but are ineffective to forecast weather accurately when there is a change in weather pattern. The shift in weather often depends on changes observed in subsequent mutually related weather variables. Using multivariate weather variables to forecast a single target weather feature can be used to determine the mutual influence and attention weight (spatial influence) of multiple weather variables with respect to target variable. The attention mechanism [32] can be used to assign different weights to input variables by determining which part of the input data needs to be focused on in the model. An attention aware LSTM model was proposed [33] to forecast soil moisture and soil temperature to perform multi-feature attention and temporal attention. The model produces an average R2 of 0.908 and 0.715 and RMSE of 1.665 and 2.756 for soil temperature and soil moisture respectively. Shi *et al.* [34] demonstrated a Self-attention joint spatiotemporal convLSTM model for temperature prediction which introduces a unified memory to define spatial and temporal models. However, the complexity and variance explained by these models are comparatively lesser.

Table 1 summarize the existing LSTM based temperature forecasting models with their limitations.

A. RESEARCH GAP

The identified research gap is to accurately forecast weather when there is a sudden change in weather patterns. The major limitations described in table 1 is that the proposed baseline and derived models forecast temperature inaccurately when there is a change observed in weather over the learned time sequence. Progressively, meteorological studies suggest that the shift in weather often depends on changes observed in subsequent mutually related weather variables. This interaction of mutually correlated weather features can be learned during weather forecasting to accurately predict a weather feature when there is a sudden change observed in weather. Thus, we aim to develop a spatial feature attention mechanism to simultaneously learn input feature interactions in long sequences to predict the target feature accurately.

III. LONG SHORT-TERM MEMORY

Traditional RNN is general form of feed forward neural network with an internal memory. The decision of the output is made by current input which is learned from the previous input and thus the output is connected to previous inputs of the sequence. It is recurrent in nature because it computes the output using the same function for every input while the output is dependent on previous calculations. They use internal state memory to process sequences of input.

Fig. 1 depicts a simple RNN where X_0 to X_n are the inputs at every sequence, H_0 to H_n are the corresponding outputs produced for every sequence. Here, we can clearly see that all the inputs are related to each other where A denotes a single RNN cell. The formula for current state, activation function and output state are described in (1), (2) and (3) respectively, where H is the single hidden vector, W is weight, W_{h-1} is the weight of previous hidden state, W_h is the weight of current input state, W_y is the weight at output state, Y_t is the output state and tanh is the activation function which regulates the values to range [−1,1].

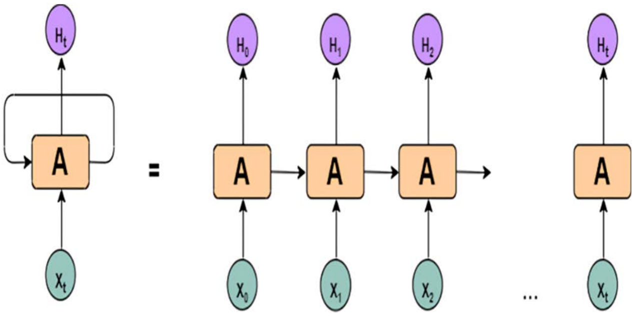

FIGURE 1. An unrolled recurrent neural network.

$$H_t = f(H_{t-1}, X_t) \tag{1}$$
$$H_t = \tanh(W_{h-1}H_{t-1} + W_h X_t) \tag{2}$$
$$Y_t = W_y H_t \tag{3}$$

LSTM [38], an artificial RNN architecture was proposed by S. Hochreiter and J. Schmidhuber in 1997. It uses a gated mechanism (input gate, output gate and forget gate) to control the flow, storage and dependency of information over time [39] thus making it well suitable for training long sequential data. LSTM was a solution to handle long term dependency, vanishing gradient and exploding problem of traditional RNNs. Fig. 2 depicts a gated LSTM network. Here, X_t and H_t denotes input and output of particular cell respectively. In the input gate, the sigmoid function regulates the information (4) and decides on the values to be remembered using H_{t-1} and X_t. The tanh function (5) assigns weights to the values passed and produces a vector V_t containing values ranging from −1 to 1.

$$i_t = \sigma\,(W_i.\,[H_{t-1}, X_t]) \tag{4}$$
$$V_t = \tanh\,(W_c.\,[H_{t-1}, X_t]) \tag{5}$$
$$f_t = \sigma\,(W_f.\,[H_{t-1}, X_t]) \tag{6}$$
$$O_t = \sigma\,(W_o.\,[H_{t-1}, X_t]) \tag{7}$$
$$H_t = O_t \otimes \tanh\,(C_t) \tag{8}$$

The output produced at input gate input gate is the element-wise product of V_t and regulated values (i_t) to produce useful information. The forget gate is responsible for discarding the information that is no longer useful. The inputs of this gate $H_{t-1}$and X_t are multiplied with weight matrix W_f and are passed through the activation function which assigns a binary value 0 or 1 to either discard or retain the information accordingly (6). C_t and memory of the block is used for extracting the useful information from output gate. Tanh function provides weights to the values which is multiplied

TABLE 1. Summary of existing LSTM based temperatue forecasting models.

	Problems Covered	Models Used	Dataset		Results	Limitation
			Period	features		
[24]	Forecast maximum and minimum temperature by exploiting local information in time-series to obtain good performance. It considers samples in the test point vicinity to have higher impact in forecasting.	Transductive-LSTM	Daily. 2007 to mid-2014 for 5 cities	Minimum Temperature, maximum temperature, dew point and wind speed	Least mean average error (MAE) 1.5 and 1.3 is obtained in predicting minimum and maximum temperature respectively for one day ahead data based on past 10 days for nov/dec test set.	In transductive learning, separate model needs to be trained for each test-point making it unsuitable for large data. It is computationally expensive and the model parameters depend on the test point feature vector.
[33]	Forecast soil temperature and soil moisture by taking predictor attention, temporal attention and multi-feature attention into account.	ILSTM_Soil	Daily data collected for 10 flux tower sites with longest duration between 1996-2014	Month of the year, day of the year, soil moisture, longwave radiation, shortwave radiation, air temperature, atmosphere pressure, wind speed, precipitation and soil temperature	Average R2, MAE and RMSE values obtained for predicting soil temperature are 0.947, 0.988 and 1.274	Model performance decreases with increase in lead time
[35]	Forecast hourly soil temperature based on daily average soil temperature (ST) and ST amplitude which is the difference between hourly ST and daily average ST	BiLSTM	Hourly data collected from 2010 - 2014	air temperature, air temperature maximum, air temperature minimum, wind speed average, solar radiation average, dew point temperature, relative humidity minimum, relative humidity maximum, vapour pressure and soil temperature	Minimum MAE and RMSE for predicting hourly ST obtained is 1.53 and 0.22 respectively. R2 of 0.923 is obtained	Results are not consistent with all the sites tested. No major architectural changes observed in the model.
[36]	Test the performance of DNN in temperature forecasting for upto 24h in comparison to SARIMA and if the model is able to predict immediate weather changes	ConvLSTM & multivariate LSTM, LSTM, SARIMA, Naïve forest	Hourly data collected between 2009 – 2013 and 2014-2018	Air temperature, wind speed, relative humidity, relative air pressure, cloud coverage, wind direction, hourly precipitation.	The average RMSE and MASE obtained by convLSTM is 2.10 and 0.93 and multivariate LSTM is 2.37 and 0.99	Not able to predict well with change in weather patterns. convLSTM performs best after 6h.
[37]	Addition of spatial information in LSTM model to improve prediction performance	Spatio-temporal stacked LSTM	Daily data from 2007 to mid-2014 for 5 cities	18 weather variables which includes temperature and humidity	Least MAE of 1.43 and 1.22 is obtained in predicting minimum and maximum temperature respectively for one day ahead data based on past 10 days for nov/dec test set.	Does not provide spatial weights obtained from prediction analysis since the spatial attention module is stacked above temporal attention module. Only captures local spatial correlations and not global correlations

with the regulated values O_t obtained from sigmoid function (7), and the resultant vector H_t (8) is the output of the cell which acts as input to the next cell. Since the proposition of original LSTM architecture, several different variations and approaches have been proposed to enhance the performance of the model such as bidirectional LSTM [40], encoder-decoder based LSTM [41] and many more [12], [42].

IV. PROPOSED WORK

We propose a novel deep learning Spatial Feature attention-based LSTM (SFA-LSTM) model to capture accurate spatial and temporal relations of multiple weather variables to forecast a weather feature. Significant spatial feature interpretations of historical data aligned directly to output feature helps the model to forecast data accurately. The model is built using encoder-decoder architecture, where the temporal dependencies in data are learnt using LSTM layers in the encoder phase and spatial feature relations in the decoder phase.

The proposed model:

- provides meaningful spatial feature interpretations which are verified using domain knowledge

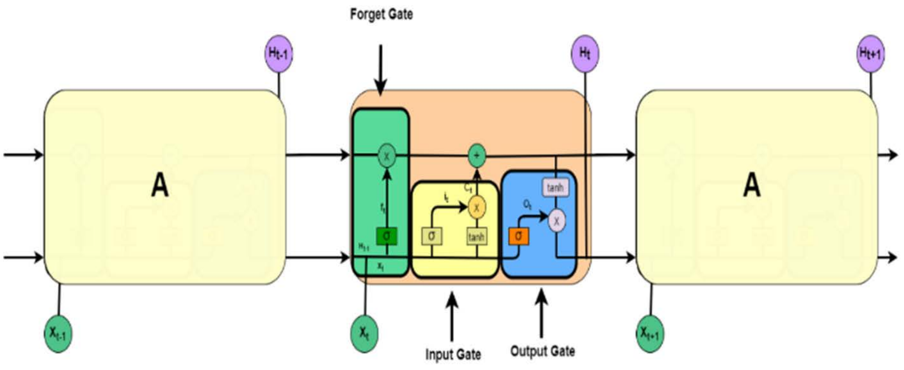

FIGURE 2. LSTM gated architecture.

- has spatial attention module built in the decoder phase to explicitly capture spatial features correlation which aligns directly with the output
- is computationally inexpensive, scalable and is dependent only on past historical data
- consists spatial feature attention and long-term temporal dependency mechanisms coordinated in a unified architecture to forecast accurately while offering precise spatial feature interpretability
- can be extended to different domain specific use cases for multivariate time series forecasting

A. SPATIAL FEATURE ATTENTION BASED LSTM (SFA-LSTM) MODEL

In this section, we propose SFA-LSTM model and investigate its computational complexity. Contrary to previous works [47], [48], [49], in SFA-LSTM, the spatial attention is designed in the decoder layer to simultaneously learn through relevant time steps and significant variables.

Our model constitutes of two major divisions which are encoder and decoder. Given a multivariate time series with N features denoted by $X = [x^1, x^2, x^3, \ldots, x^N]^T \in R^{NXTin}$, where T_{in} is the total length of input sequence and $x^i = [x_1^i, x_2^i, x_3^i, \ldots, x_{Tin}^i]^T \in R^{Tin}$, $i \in [1, N]$ indicates time series associated with each input feature. To represent all input features at time step $t \in [1, Tin]$ such that $X = [x_1, x_2, \ldots, x_{Tin}]^T$ is the compact form of all input time series then we denote it as $x_t = [x_t^1, x_t^2, x_t^3, \ldots, x_t^N]^T \in R^N$. Analogously, the output univariate time series for $\mathrm{T_{out}}$time steps is denoted by y $\in \mathrm{R^{Tout}}$, where $y_j \in R$ is the output at time step j.

In our model, for every time step t, temporal dependencies and spatial feature attention are calculated. The input to the encoder at a time step t is $x_t = [x_t^1, x_t^2, x_t^3, \ldots, x_t^N]^T \in R^N$. The encoder consists of an LSTM layer which calculates temporal hidden states and dependencies for input time series $X = [x_1, x_2, \ldots, x_{Tin}]^T$. The LSTM layer reads the input sequences from x_1 *to* x_{Tin} and generates a sequence of hidden states at the encoder, represented as $H = [h_1, h_2, \ldots, h_{Tin}]^T$, where $h_t \in R^Q$. The hidden states produced at the encoder for every time step t acts as temporal embeddings for the decoder temporal LSTM layer ($LSTM_t$). h_t' and c_t' are hidden state and cell state of $LSTM_t$.

The spatial feature attention module is built in the decoder parallel to temporal layer to capture spatial feature correlations while attending most relevant time steps as it directly aligns with the output feature as shown in fig. 3. Spatial feature embeddings are generated independently using feed forward neural network which are input to the spatial feature attention module. The feedforward neural network used to compute spatial feature embeddings include a series of computations where the data from previous hidden state of decoder is concatenated with input features acted upon by soft-max activation function to assign weights in the decoder LSTM. The spatial feature embeddings do not have any feedback connections i.e., for each feature $x^i = [x_1^i, x_2^i, x_3^i, \ldots, x_{Tin}^i]^T \in R^{Tin}$, $i \in [1, N]$, the spatial embeddings for all features are computed from $X = [x^1, x^2, x^3, \ldots, x^N]^T$ and denoted as $S = [s^1, s^2, s^3, \ldots, s^N]^T$. The spatial attention weights are calculated in a feed forward aligned manner in the decoder layer where h_f' and c_f' are hidden state and cell state of spatial feature attention. β_j^i is the spatial attention weight of i^{th} feature calculated at output time step j using $h_{f,j-1}'$ which is previous hidden state of spatial feature attention at the decoder and $\mathrm{s_i}$ is the spatial feature embedding of i^{th} feature. $W_\alpha \in R^P+Q$ is the learning parameter and tanh activation function simulates the weights to the values passed (10).

$$d_j^i = tanh\left(W_\alpha^T\left[h_{f,j-1}', s^i\right]\right) \tag{9}$$

$$\beta_j^i = \frac{exp\left(d_j^i\right)}{\sum_1^N exp\left(d_j^0\right)}, f_j = \sum\nolimits_1^N \beta_j^i s^i \tag{10}$$

We then use spatial feature attention weights to calculate spatial feature context vector f_j, and it is distinct at each time step. f_j is further optimized and its dimension is reduced using feed forward neural network with tanh activation to produce $r_{f,j}$ which is further concatenated with the output of previous time step O_{j-1}. This produce an updated spatial

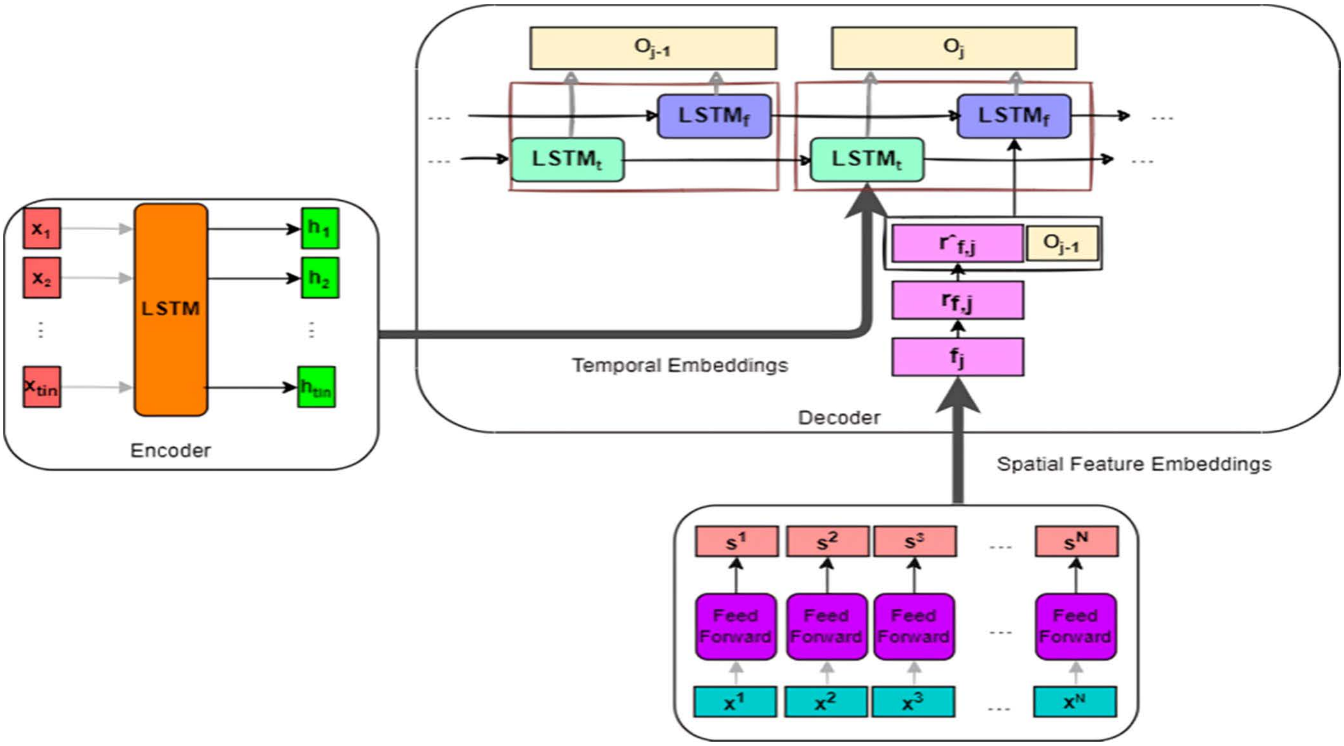

FIGURE 3. SFA-LSTM working architecture to compute output O_j at time step t.

feature context vector $\hat{r}_{f,j}$ which is input to $LSTM_f$, the final spatial feature attention LSTM layer.

$$r_{f,j} = tanh\left(W_f f_j\right), \hat{r_{f,j}} = \left[r_{f,j}, O_{j-1}\right] \tag{11}$$

$$O_j = \left[h'_{t,j}, h'_{f,j}\right] \tag{12}$$

The final step of SFA-LSTM is to concatenate the hidden states of $LSTM_t$ and $LSTM_f$ ($[h'_{t,j}, h'_{f,j}]$) which is the output O_j and append to the output list of predictions.

V. EXPERIMIENTS

Fig. 4 describes the detailed workflow of modelling. We start our implementation from the data collection and data preprocessing phase which is described in detail in the next section. Our next phase includes model setting, training, comparison and evaluation. The final step of our experimentation is to compare the performance of trained models and verify the obtained spatial feature attention weights with domain knowledge.

A. DATASET DESCRIPTION AND DATA PREPROCESSING

In this study, we use real meteorological data of weather station at Saskatoon John G. Diefenbaker Intl. Airport latitude 52.14, longitude -106.69 collected from weatherstats website. The datasets is an hourly time-series of 87672 data points each from 2012-01-01 00:00:00 CST to 2021-12-31 23:00:00 containing weather variables as temperature, dew point, windchill, relative humidity, station pressure, sea pressure and wind speed. Temperature is recorded in Celsius scale. Dew point is also in Celsius scale which provides the average temperature below which water droplets begin to condense. Relative humidity provides the fraction of water vapour present in the air. Wind speed is measured in m/s expressing the velocity of wind and surface pressure is measured in Pascals (Pa). These meteorological features are selected to forecast weather because these features explain the state of weather for a given location and time. All the eight meteorological features are used as input features to forecast temperature [24,33,35,36,37].

The experiment was performed on real data and thus included some necessary preprocessing steps to reflect true model performance. The missing values in the data were imputed using linear interpolation in forward direction. Linear interpolation estimates the missing values in the increasing order from previous values. Smoothing of data using simple moving average with an appropriate window length is an effective technique in time series forecasting as it removes noise and random variations from data without neglecting the weather variations in time. For our study, we perform simple moving average of window length = 5. Data smoothened over a higher window length might not represent the actual nature of weather. In the final stage of data preprocessing, we normalize our data using MinMax Scaling Technique. Since the proposed model is of multi-input and single-output form and our multiple input time series are in different unit and range, we normalize it in the range 0 to 1 using the

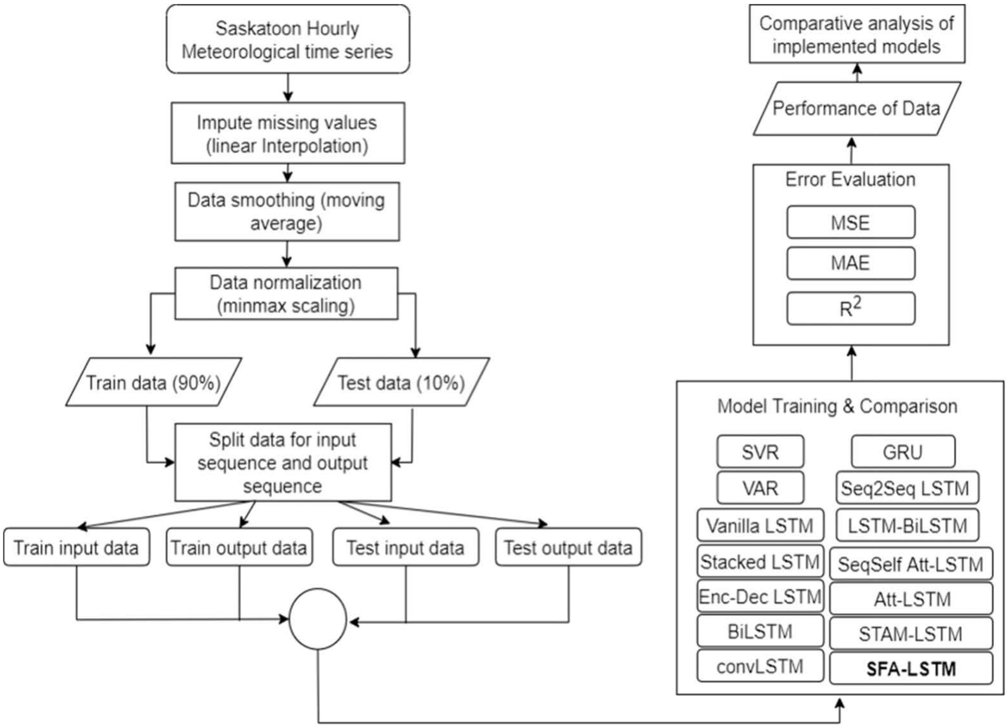
Saskatoon Hourly Meteorological time series
Impute missing values (linear Interpolation)
Data smoothing (moving average)
Data normalization (minmax scaling)
Train data (90%)
Test data (10%)
Split data for input sequence and output sequence
Train input data
Train output data
Test input data
Test output data
Model Training & Comparison
SVR
GRU
VAR
Seq2Seq LSTM
Vanilla LSTM
LSTM-BiLSTM
Stacked LSTM
SeqSelf Att-LSTM
Enc-Dec LSTM
Att-LSTM
BiLSTM
STAM-LSTM
convLSTM
SFA-LSTM
Error Evaluation
MSE
MAE
R2
Performance of Data
Comparative analysis of implemented models

FIGURE 4. Detailed workflow of modelling.

equation 13 where Xi represents the ith data point in time series from [0, n] and Xmin and Xmax represent the minimum and the maximum data point in the sequence respectively.

$$X_i=\frac{X_i-X_{min}}{X_{max}-X_{min}} \tag{13}$$

The data is split into training set and testing set of proportion 0.9 and 0.1 respectively. The training dataset contains 78877 rows and the testing data contains 8743 rows. Both training and test sets are processed using moving window algorithm to obtain the input and output sequences. The input sequence contains seven features i.e. temperature, dew point, windchill, relative humidity, station pressure, sea pressure and wind speed and the output sequence contains temperature values. We compare the performance of SFA-LSTM with several baselines and derived models which will be discussed in the next sections.

B. MODEL SETTING AND TRAINING

We applied the processed data containing seven weather variables described above to predict temperature and used tensorflow backend in our experiments. Input variables to SFA-LSTM and other studies models are temperature, dew point, windchill, relative humidity, station pressure, sea pressure and wind speed and the output variable (target variable) is temperature. LSTM is an artificial RNN with feedback connections which enables it to process long sequences. Hyperparameters are the values which need to be chosen or predefined before the training of algorithm. These hyperparameters are the not the parameters of machine learning that will be learned during the training of model. The hyperparameters of LSTM include learning rate, hidden states, batch size, epochs and optimizer. The working evaluation mechanism of hyper parameter tuning is depicted in Fig. 5.

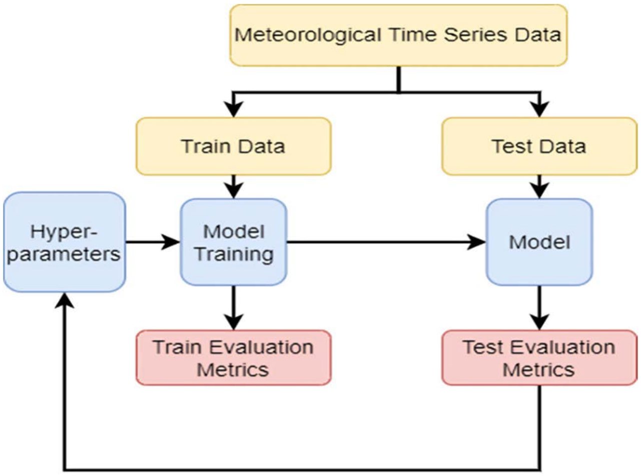

FIGURE 5. Evaluation mechanism in hyperparameter tuning.

We chose Bayesian Optimizer model to tune the hyperparameters of SFA-LSTM. This is to keep a track of past evaluation results which will be used in probabilistic algorithm of Bayesian algorithm. Learning rate decides how fast the model will converge or diverge or in other words, it decides on how quickly the learning parameters of the model are updated. If a higher learning rate is set then the model may not converge and produce biased results and if a lower learning rate is set then it will drastically slow down the learning. We train our model three times for learning rates 0.01, 0.001 and 0.0001 and learning rate 0.0001 results to the minimum loss of 8.413637260673568e-06. Hidden states in deep learning decide the capacity of the model to learn. It is the main measure of learning capacity of the deep learning model. A thumb rule is that the more the complex model is, the more hidden units/states it requires to learn.

We train our model for 16, 32, 64 and 128 hidden states using Bayesian optimizer and the choose 32 for final model training. Batch size of model defines the number of resources allocated for model training and the speed of model. Defining a higher batch size for model training is computationally expensive and a smaller batch size will induce noise in the model. Thus, we train our model for batch size 128, 256 and 512. Bayesian optimizer produces the output in favor of batch size of 128. The value of epoch decides on the number of complete iterations of the data and model to be run. The value can be anything until infinity and the optimal value decides on how well the model fits the data. A smaller value for epochs will result in higher error loss and a bigger value may result to overfitting. We trained our model for 1 to 50 epochs and the results produced are shown in fig. 6. The model results in low MSE in range of e-05 after 15 epochs and we choose the size of epochs to be 20.

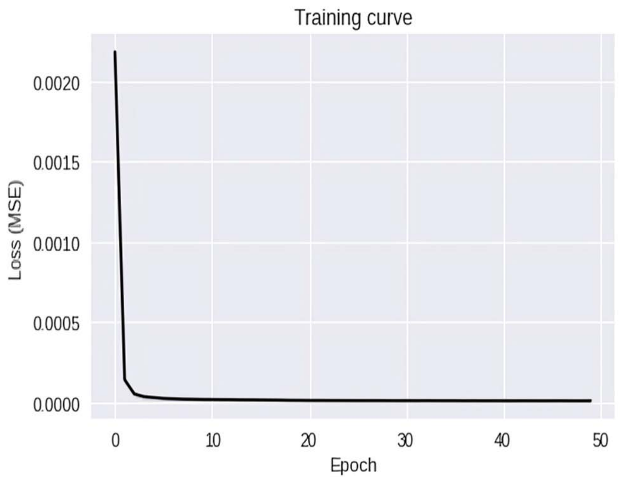

FIGURE 6. Training curve for epochs vs loss for 50 epochs.

The hyperparameters of SFA-LSTM are determined using Bayesian optimization technique i.e, learning rate of the model is 0.001, number of epochs is 50, optimizer is Adam and the activation function used is Tanh.

C. MODEL EVALUATION

We use three metrics i.e., Mean Sqaure Error (MSE), Mean Absolute Error (MAE) and R^2 to evaluate to the performance of SFA-LSTM and other state-of-the-art predictive models. MSE is the squared error loss corresponding to expected value and MAE is the average absolute error loss in a set of predictions. R^2 describes the magnitude proportion of variance explained by the predictive model. The performance metrics are calculated as follows:

$$MSE = \frac{\sum_1^N (y_i - y_i')^2}{N} \tag{14}$$

$$MAE = \frac{\sum_1^N |y_i - y_i'|}{N} \tag{15}$$

$$R^2 = 1 - \frac{\sum_1^N (y_i - y_i')^2}{\sum_1^N (y_i - y_{avg})^2} \tag{16}$$

where y_i is the actual temperature value at time step i, y_i' is the predicted temperature value at i^{th} time step, y_{avg} is the mean of actual temperature values and N is the sample size. These error scores are used as common performance metrics for regression models [50], [51], [52].

D. MODEL COMPARISON

The performance of the proposed model SFA-LSTM is compared with several baseline and derived models with state-of-the-art model results. We compared SFA-LSTM with baseline models such as Support Vector Regression (SVR), Gated Recurrent Unit (GRU), Vanilla LSTM, Stacked LSTM, Encoder-Decoder LSTM (Enc-Dec LSTM), 1D-convolutional LSTM (convLSTM), Bi-directional LSTM (Bi-LSTM) and derived models which are - Sequence to Sequence LSTM (Seq2seq-LSTM), Attention LSTM (Att-LSTM), Sequence Self-Attention LSTM (SeqSelf Att-LSTM), LSTM-BiLSTM and Spatiotemporal Attention LSTM (STAM LSTM) which has been widely applied to predict temperature values.

The hyperparameters of the implemented models is described in table 2. GRU is another variation of RNN developed by in 2014 [41]. Its performance in learning long sequences is similar to LSTM and is computationally less expensive than LSTM because of fewer gates. GRU is widely used in weather prediction modelling [43], [44], [45]. We also compare the performance of SFA-LSTM with the original LSTM model (Vanilla LSTM). We implemented an integrated LSTM-BiLSTM model which was proposed by Maddu *et al.* [35] to forecast soil temperature with multivariate input variables. Sequence to Sequence LSTM (seq2seq LSTM) model was proposed by Zaytar *et al.* [29] to forecast temperature with temperature, wind speed and relative humidity as input features. STAM-LSTM is a novel state-of-the-art spatiotemporal attention- based LSTM model proposed by Gangopadhyay *et al.* [46] for multivariate time series prediction. We use Keras Self-Attention package to implement attention mechanism in LSTM model which considers context of each time step. Additionally, we also build a custom temporal attention-based LSTM model (att-LSTM) to compare its performance with the proposed SFA-LSTM model. The performance SFA-LSTM is also compared the

TABLE 2. Proposed models hyperparameters details for comparative analysis.

Model	Hidden States	Activation Function	Hidden layers	Learning Rate	Dropout	Optimizer	Batch Size	Epochs	Additional
GRU	16	ReLu	-	0.01	-	Adam	32	20	-
Vanilla LSTM	16	ReLu	-	0.01	-	Adam	32	20	-
Stacked LSTM	16	ReLu	2 LSTM	0.01	-	Adam	32	20	-
Enc-Dec LSTM	16	ReLu	-	0.01	-	Adam	32	20	Output Layer: Time Distributed
ConvLSTM	16	ReLu	-	0.01	-	Adam	32	20	Conv1DFilter: 64, Kernel Size: 2
BiLSTM	16	ReLu	-	0.01	-	Adam	32	20	-
Seq2seq LSTM	100	ReLu	2 LSTM &1 dense	0.01	-	RMSProp	32	20	-
Att-LSTM	16	ReLu	1 LSTM	0.01	-	Adam	32	20	-
Seq Self-Att LSTM	16	ReLu	1 LSTM	0.1	-	Adam	32	20	Attention Activation: Sigmoid
LSTM-BiLSTM	14,14,14,14,6	ReLu	3 LSTM & 4 alt. dropout	0.01	0.20	Adam	-	10	-
STAM-LSTM	32	ReLu	-	0.01	0.20	-	256	10	Temporal Dimensionality Reduction: 4
SFA-LSTM	32	Tanh	1 LSTM	0.0001	-	Adam	128	25	Spatial Feature Attention Module

TABLE 3. Feature comparisons between existing and the proposed SFA-LSTM model.

Method	Com. In.	Com. Sc.	Hist. Dep.	Sp. Feature Int.	Hidden States	Activation Function	Hidden layers	Learning Rate	Dropout	Optimizer	Batch Size	Epochs	Additional
Transductive LSTM [24]	×	×	×	×	32	Tanh	1 LSTM	0.0001	-	Adam	128	25	Transductive cosine similarity
Integrated Bi-LSTM [35]	×	✓	✓	×	-	Tanh	-	-	-	Adam	-	-	-
Multivariate convLSTM [36]	×	✓	✓	×	14,14,14,14,6	ReLu	3 LSTM & 4 alt. dropout	0.01	0.20	Adam	-	10	-
ILSTM [33]	×	✓	✓	×	-	Tanh	6 convolutional	-	-	-	-	1000	Multi-feature attention
Spatio-Temporal Stacked LSTM [37]	×	✓	✓	×	-	Tanh	-	0.001	-	-	64	30	-
Our proposed SFA-LSTM	✓	✓	✓	✓	**32**	**Tanh**	**3 LSTM**	**0.0001**	**-**	**Adam**	**128**	**-25**	Spatial Feature Attention Module

Com. In[1].: Computationally Inexpensive, Com. Sc.[2]: Computationally Scalable, Hist. Dep.[3] Historical Dependency, Sp. Feature Int[4].: Spatial Feature Interpretability.

algorithms and their performances obtained in literature. The information has been tabulated in table 3 with feature comparison.

VI. RESULTS

A novel deep learning model, SFA-LSTM for short term weather forecasting has been proposed in this research. The proposed model is evaluated using statistical error metrics i.e, MAE, MSE and R^2 and its performance is compared with baseline, derived and existing models surveyed in literature. The results will also include an analysis on the spatial feature interpretability and the spatial feature attention weights obtained during model learning with its verification using domain knowledge.

A. SHORT TERM TEMPERATURE PREDICTION AND MODEL COMPARISON

Table 4 contains the quantitative findings and prediction performance of our proposed algorithms listed in table 2. These models are trained and developed by us using the hyperparameters described in Table 2 with input sequence of 24hr and output sequence of 1hr. This means that 1hr temperature is predicted based on past 24hr meteorological values. The performance of SFA-LSTM which outperforms other proposed models is also compared with the results of existing models from literature (feature comparison with our proposed SFA-LSTM in Table 3) and the same has been documented in Table 5 and visually depicted in fig 7.

TABLE 4. Empirical results for 1hr ahead prediction with input sequence of past 24hr.

Model	MSE	MAE	R2 Score
SVR	14.508	3.43560	0.85810
GRU	0.13154	0.2888	0.9994
Vanilla LSTM	0.15489	0.31904	0.99932
Stacked LSTM	0.1106	0.25516	0.99951
Enc-Dec LSTM	0.13978	0.29945	0.9993
ConvLSTM	0.51771	0.55076	0.99774
BiLSTM	0.13645	0.29561	0.99940
Seq2SeqLSTM	0.2015	0.37154	0.99912
Att-LSTM	0.17042	0.34381	0.99925
SeqSelfAtt-LSTM	0.17564	0.31601	0.99923
LSTM-BiLSTM	0.91977	0.7070	0.99599
STAM-LSTM	1.3746	0.9028	0.9940
SFA-LSTM	**0.0871**	**0.2317**	**0.9996**

TABLE 5. Model performance comparison based on error scores listed in literature.

Model	MSE	MAE
Our proposed (best of Table IV) SFA-LSTM	**0.0871**	**0.2317**
Transductive LSTM [24]	3.74	1.50
Integrated Bi-LSTM [35]	3.01	1.33
Multivariate convLSTM [36]	0.1661	0.2797
ILSTM [33]	0.558	0.608
Spatio-Temporal Stacked LSTM [37]	3.64	1.43

Clearly, our proposed model SFA-LSTM out performs the baseline and derived model as well as the existing models with MSE of 0.0871 and MAE of 0.2317 explaining 99% variance of our data. The prediction output is depicted in fig 8 for 48hr ahead data in testing phase, in fig 9 for 96hr ahead data in testing phase and in fig 10 for the whole testing data. The interactions of various correlated meteorological input features help to predict temperature accurately when there is a change in weather pattern over the sequence.

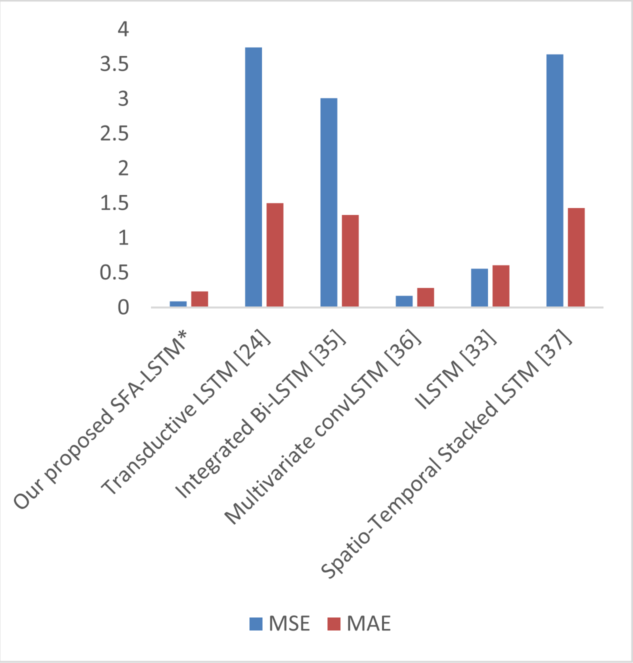

FIGURE 7. Performance of proposed model compared with models listed in literature based on MSE and MAE.

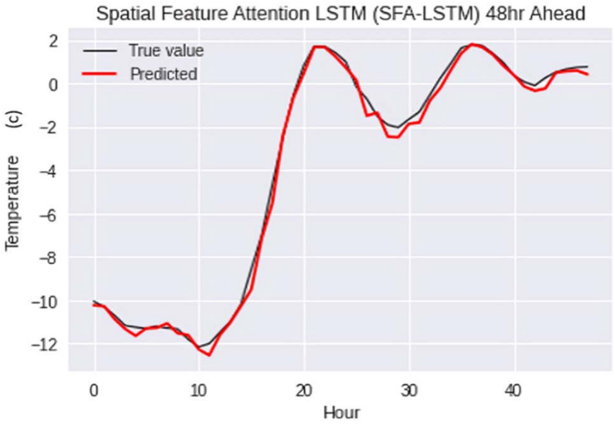

FIGURE 8. Temperature predicted vs temperature observed by SFA-LSTM for 48hr ahead.

B. SHORT TERM TEMPERATURE PREDICTION FOR DIFFERENT INPUT SEQUENCES AND OUTPUT SEQUENCES

The performance of SFA-LSTM for various input and output sequence lengths is documented in Table 6. On comparison, we can safely say that SFA-LSTM has better prediction accuracy as compared to other models for different input and output sequence lengths.

C. SPATIAL FEATURE INTERPRETABILITY

Table 7 provides the correlation between input features used for temperature prediction. Correlation is a statistical value to measure the amount of linear dependency between two variables. The use of this information in temperature prediction will help us to understand the spatial feature interpretability

TABLE 6. Temperature predicted with 24hr, 48hr and 72hr input sequence for 1hr, 2hr and 3hr ahead.

Seq Len	Steps ahead	Mean Square Error (MSE)				Mean Absolute Error (MAE)			
		GRU	LSTM	Bi-LSTM	SFA-LSTM	GRU	LSTM	Bi-LSTM	SFA-LSTM
24	1	0.118	0.157	0.620	0.0640	0.262	0.315	0.656	0.1894
	2	0.220	0.241	0.646	0.1824	0.334	0.355	0.630	0.3054
	3	0.560	0.509	1.430	0.3867	0.525	0.499	0.930	0.4323
48	1	0.275	0.0738	0.330	0.0701	0.451	0.201	0.459	0.1975
	2	0.291	0.363	0.764	0.2001	0.383	0.465	0.686	0.3175
	3	0.767	0.5009	0.8417	0.683	0.628	0.5211	0.668	0.6790
72	1	0.127	0.098	0.625	0.0518	0.271	0.240	0.606	0.1682
	2	0.495	0.333	0.640	0.1820	0.552	0.426	0.618	0.3104
	3	0.672	0.541	0.9704	0.364	0.613	0.5271	0.7013	0.4174

TABLE 7. Correlation matrix of feature set.

Feature	Temperature	Dew point	Wind chill	Relative humidity	Pressure Station	Pressure sea	Wind speed
Temperature	1	0.9291	0.8608	-0.4617	-0.2945	-0.4743	0.0987
Dew point	0.9291	1	0.8783	-0.1271	-0.3443	-0.5179	0.0271
Wind chill	0.8608	0.8783	1	-0.1980	-0.3353	-0.4896	0.0044
Relative humidity	-0.4617	-0.1271	-0.1980	1	-0.0521	0.0168	-0.2280
Pressure station	-0.2945	-0.3443	-0.3353	-0.0521	1	0.9775	-0.2156
Pressure sea	-0.4743	-0.5179	-0.4896	0.0168	0.9775	1	-0.2086
Wind speed	0.0987	0.0271	0.0044	-0.2280	-0.2156	-0.2086	1

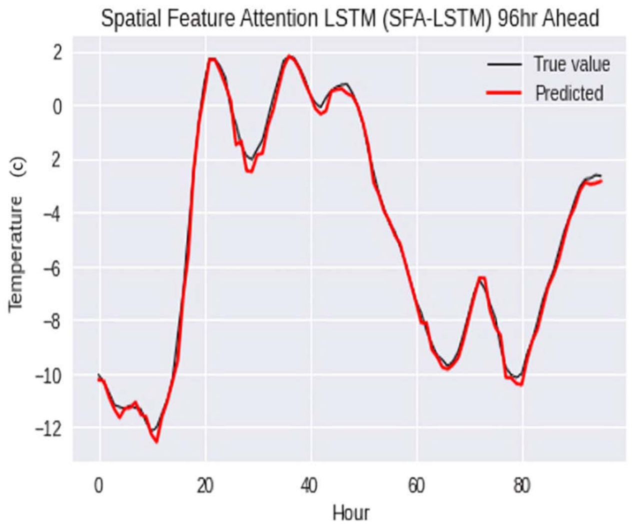

FIGURE 9. Temperature predicted vs temperature observed by SFA-LSTM for 96hr Ahead.

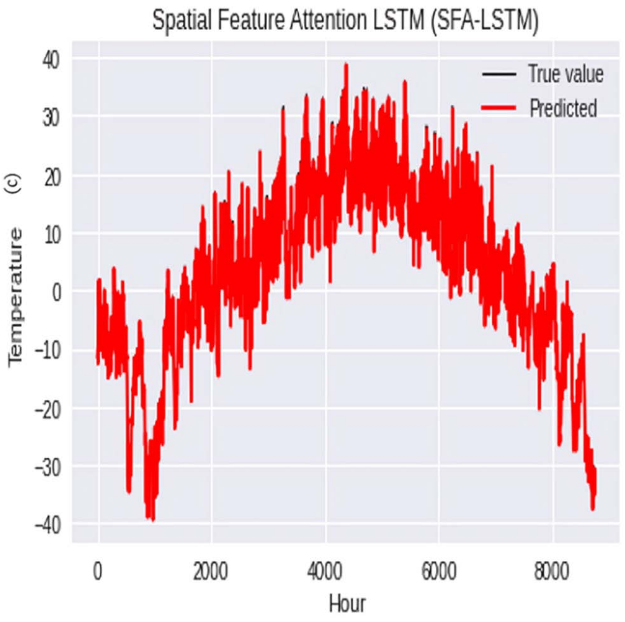

FIGURE 10. Temperature predicted vs temperature observed by SFA-LSTM for entire test set (8743hr).

between the input feature and the target feature. The spatial feature attention weights obtained from learning the SFA-LSTM model is depicted in fig. 11.

Clearly, temperature contributes maximum towards forecasting future temperature values i.e., upto 20% of total feature contribution. Dew point contributes upto 19% towards temperature prediction and is linearly correlated to a great extent. Wind speed is seen to contribute the least with only 0.85% of total contribution and is also correlated to temperature to a very small extent.

We observe that the spatial feature attention weights obtained from SFA-LSTM are verified using domain knowledge. The spatial feature attention mechanism helps to forecast weather accurately when there is a change in weather values over the sequence.

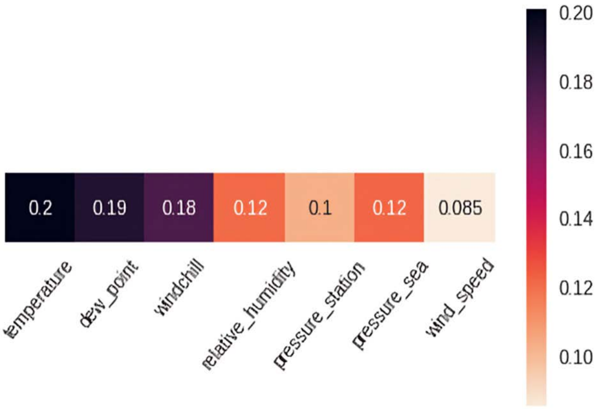

FIGURE 11. Spatial feature attention weights.

VII. CONCLUSION

In this work, weather forecasting problem is addressed with the vision to accurately forecast weather when a sudden change in weather pattern is observed. To address this problem, we used the concept of mutual correlation between meteorological features. In this paper, we proposed our novel SFA-LSTM model with a built-in spatial feature attention mechanism to capture long term dependencies and spatial feature correlations of multivariate input time series to predict a single output feature. The spatial feature attention mechanism grasps the quantitative mutual influence of input features on target feature which leads to accurate predictions including when sudden changes in input sequences are observed.

The magnitude of shift in a weather feature can be learned from simultaneous shifts observed in subsequent mutually related weather variables. Using multivariate weather variables to forecast a single target weather feature can be used to determine the weight of spatial feature influence of multiple weather variable on the target variable. Capturing such correlations during model learning helps to predict future weather accurately over long sequences. The proposed model was built using encoder-decoder architecture, where the temporal dependencies in data are learnt using LSTM layers in the encoder phase and spatial relations in the decoder phase. SFA-LSTM is seen to outperform the state-of-the-art model performance with providing accurate spatial feature interpretability.

REFERENCES

[1] F. V. Davenport and N. S. Diffenbaugh, "Using machine learning to analyze physical causes of climate change: A case study of U.S. Midwest extreme precipitation," *Geophys. Res. Lett.*, vol. 48, no. 15, Aug. 2021, Art. no. e2021GL093787.

[2] J. Xie, J. Zhang, J. Yu, and L. Xu, "An adaptive scale sea surface temperature predicting method based on deep learning with attention mechanism," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 5, pp. 740–744, May 2020.

[3] N. Singh, S. Chaturvedi, and S. Akhter, "Weather forecasting using machine learning algorithm," in *Proc. Int. Conf. Signal Process. Commun. (ICSC)*, Mar. 2019, pp. 171–174.

[4] B. Wang, J. Lu, Z. Yan, H. Luo, T. Li, Y. Zheng, and G. Zhang, "Deep uncertainty quantification: A machine learning approach for weather forecasting," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2087–2095.

[5] R. I. Rasel, N. Sultana, and P. Meesad, "An application of data mining and machine learning for weather forecasting," in *Advances in Intelligent Systems and Computing*. Cham, Switzerland: Springer, 2018, pp. 169–178.

[6] O. Shahid, M. Nasajpour, S. Pouriyeh, R. M. Parizi, M. Han, M. Valero, F. Li, M. Aledhari, and Q. Z. Sheng, "Machine learning research towards combating COVID-19: Virus detection, spread prevention, and medical assistance," *J. Biomed. Informat.*, vol. 117, May 2021, Art. no. 103751.

[7] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104340.

[8] I. Maqsood, M. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Comput. Appl.*, vol. 13, no. 2, pp. 112–122, Jun. 2004.

[9] M. Holmstrom, D. Liu, and C. Vo. (2016). *learning Applied to Weather Forecasting*. Stanford. Accessed: May 19, 2022. [Online]. Available: http://cs229.stanford.edu/proj2016/report/HolmstromLiuVo-MachineLearningAppliedToWeatherForecasting-report.pdf

[10] L. Cornejo-Bueno, L. Cuadra, S. Jiménez-Fernández, J. Acevedo-Rodríguez, L. Prieto, and S. Salcedo-Sanz, "Wind power ramp events prediction with hybrid machine learning regression techniques and reanalysis data," *Energies*, vol. 10, no. 11, p. 1784, 2017.

[11] A. H M Jakaria, M. M. Hossain, and M. A. Rahman, "Smart weather forecasting using machine learning: A case study in Tennessee," 2020, *arXiv:2008.10789*.

[12] Edu.cn. *Support Vector Machines for Classification and Regression*. Accessed: May 19, 2022. [Online]. Available: https://see.xidian.edu.cn/faculty/chzheng/bishe/indexfiles/new_folder/SVM.pdf

[13] C. Gershenson, "Artificial neural networks for beginners," Tech. Rep., 2003. [Online]. Available: https://arxiv.org/abs/cs/0308031

[14] A. G. Salman, B. Kanigoro, and Y. Heryadi, "Weather forecasting using deep learning techniques," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Oct. 2015, pp. 281–285.

[15] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, 2006, pp. 1–8.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] S. Hu, Y. Xiang, D. Huo, S. Jawad, and J. Liu, "An improved deep belief network based hybrid forecasting method for wind power," *Energy*, vol. 224, Jun. 2021, Art. no. 120185.

[18] Y. Cheng, X. Zhou, S. Wan, and K.-K.-R. Choo, "Deep belief network for meteorological time series prediction in the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4369–4376, Jun. 2019.

[19] L. Medsker and L. C. Jain, *Recurrent Neural Networks: Design and Applications*. Boca Raton, FL, USA: CRC Press, 1999.

[20] W. Fang, Y. Chen, and Q. Xue, "Survey on research of RNN-based spatio-temporal sequence prediction algorithms," *J. Big Data*, vol. 3, no. 3, pp. 97–110, 2021.

[21] P. Le and W. Zuidema, "Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs," 2016, *arXiv:1603.00423*.

[22] N. K. Manaswi, *Deep Learning With Applications Using Python*. Berkeley, CA, USA: Apress, 2018.

[23] R. K. Agrawal, F. Muchahary, and M. M. Tripathi, "Long term load forecasting with hourly predictions based on long-short-term-memory networks," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2018, pp. 1–6.

[24] Z. Karevan and J. A. K. Suykens, "Transductive LSTM for time-series prediction: An application to weather forecasting," *Neural Netw.*, vol. 125, pp. 1–9, May 2020.

[25] A. Milad, I. Adwan, S. A. Majeed, N. I. M. Yusoff, N. Al-Ansari, and Z. M. Yaseen, "Emerging technologies of deep learning models development for pavement temperature prediction," *IEEE Access*, vol. 9, pp. 23840–23849, 2021.

[26] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[27] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Comput.*, vol. 24, no. 21, pp. 16453–16482, Nov. 2020.

[28] D. Kreuzer, M. Munz, and S. Schlüter, "Short-term temperature forecasts using a convolutional neural network—An application to different weather stations in Germany," *Mach. Learn. with Appl.*, vol. 2, Dec. 2020, Art. no. 100007.

[29] M. Akram and C. El, "Sequence to sequence weather forecasting with long short-term memory recurrent neural networks," *Int. J. Comput. Appl.*, vol. 143, no. 11, pp. 7–11, 2016.

[30] I. Park, H. S. Kim, J. Lee, J. H. Kim, C. H. Song, and H. K. Kim, "Temperature prediction using the missing data refinement model based on a long short-term memory neural network," *Atmosphere*, vol. 10, no. 11, p. 718, Nov. 2019.

[31] D. S. Roy, "Forecasting the air temperature at a weather station using deep neural networks," *Proc. Comput. Sci.*, vol. 178, pp. 38–46, Jan. 2020.

[32] J. Newman and B. J. Baars, "A neural attentional model for access to consciousness: A global workspace perspective," *Concepts Neurosci.*, vol. 4, no. 2, pp. 255–290, 1993.

[33] Q. Li, Y. Zhu, W. Shangguan, X. Wang, L. Li, and F. Yu, "An attention-aware LSTM model for soil moisture and soil temperature prediction," *Geoderma*, vol. 409, Mar. 2022, Art. no. 115651.

[34] L. Shi, N. Liang, X. Xu, T. Li, and Z. Zhang, "SA-JSTN: Self-attention joint spatiotemporal network for temperature forecasting," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9475–9485, 2021.

[35] R. Maddu, A. R. Vanga, J. K. Sajja, G. Basha, and R. Shaik, "Prediction of land surface temperature of major coastal cities of India using bidirectional LSTM neural networks," *J. Water Climate Change*, vol. 12, no. 8, pp. 3801–3819, Dec. 2021.

[36] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*.

[37] Z. Karevan and J. A. K. Suykens, "Spatio-temporal stacked LSTM for temperature prediction in weather forecasting," 2018, *arXiv:1811.06341*.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112896.

[40] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.

[41] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[42] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. *Learning Precise Timing With LSTM Recurrent Networks*. Accessed: May 19, 2022. [Online]. Available: https://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf

[43] M. Chhetri, S. Kumar, P. Pratim Roy, and B.-G. Kim, "Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan," *Remote Sens.*, vol. 12, no. 19, p. 3174, Sep. 2020.

[44] J. M. Han, Y. Q. Ang, A. Malkawi, and H. W. Samuelson, "Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements," *Building Environ.*, vol. 192, Apr. 2021, Art. no. 107601.

[45] D. Sun, J. Wu, H. Huang, R. Wang, F. Liang, and H. Xinhua, "Prediction of short-time rainfall based on deep learning," *Math. Problems Eng.*, vol. 2021, pp. 1–8, Mar. 2021.

[46] T. Gangopadhyay, S. Y. Tan, Z. Jiang, R. Meng, and S. Sarkar, "Spatiotemporal attention for multivariate time series prediction and interpretation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3560–3564.

[47] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," 2017, *arXiv:1704.02971*.

[48] Y. Liu, C. Gong, L. Yang, and Y. Chen, "DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113082.

[49] J. Hu and W. Zheng, "Multistage attention network for multivariate time series prediction," *Neurocomputing*, vol. 383, pp. 122–137, Mar. 2020.

[50] P. Hewage, A. Behera, M. Trovati, and E. Pereira, "Long-short term memory for an effective short-term weather forecasting model using surface weather data," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.*, 2019, pp. 382–390.

[51] S. S. Baboo and I. K. Shereef, "An efficient weather forecasting system using artificial neural network," *Int. J. Environ. Sci. Develop.*, vol. 1, no. 4, p. 321, 2010.

[52] R. Castro, Y. M. Souto, E. Ogasawara, F. Porto, and E. Bezerra, "STConvS2S: Spatiotemporal convolutional sequence to sequence network for weather forecasting," *Neurocomputing*, vol. 426, pp. 285–298, Feb. 2021.

MASOOMA ALI RAZA SULEMAN is currently pursuing the bachelor's degree in computer science and engineering with the Vellore Institute of Technology, Chennai, India. She is also an International Visiting Research Trainee at York University. Her research interests include machine learning and deep learning in sustainable environmental development and building data science tools for environmental modeling.

S. SHRIDEVI received the Doctorate degree in computer science from Manonmaniam Sundaranar University (MSU), Tirunelveli. She is currently working as an Associate Professor with the Centre for Advanced Data Science, Vellore Institute of Technology, Chennai. She was a University Rank Holder and a Medalist in her master's degree. She has published many patents and research papers in reputed international journals and conferences and has received best research awards for her works. Her research interests include semantic artificial intelligence, web mining, machine learning, and deep learning. She received seed funds for her research project and had applied to funds for many government agencies as well.