SCIENTIFIC CULTURE

www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 8078-8092
Open Access. Online & Print

***DOI**:https://doi.org/10.5281/zenodo.20520593*

HYBRID CNN-LSTM MODELS FOR MEDIUM-TERM SPI-3 AND SPI-6 DROUGHT FORECASTING IN TEL RIVER BASIN, ODISHA, INDIA

Abhisek Kar[1]*, Sujata Chakravarty[2], Bibhuti Bhusan Sahoo[3]

[1]*PhD Research Scholar, Department of Computer Science & Engineering, Centurion University of Technology and Management, Odisha, India-752050.*

[2] *Professor, Department of Computer Science & Engineering, Centurion University of Technology and Management, Odisha, India. Email: sujata. Chakravarty@cutm.ac.in*

[3]*Associate Professor, Department of Agricultural Engineering, Centurion University of Technology and Management, Odisha, India-761112. Email: bibhuti.sahoo@cutm.ac.in*

Received: **27/11/2025**
Accepted: **06/04/2026**

Corresponding Author: Abhisek Kar (okar47889@gmail.com)

ABSTRACT

Drought is a devastating hydroclimatic hazard, with impacts on agriculture, water resources and human lives. Accurate forecasts are crucial for managing drought risks and issuing early warnings. We use deep learning models to forecast medium-term meteorological droughts for the Tel River Basin, a drought-prone sub-basin of the Mahanadi River in Odisha, India. Three- and six-month Standardized Precipitation Index (SPI-3, SPI-6) time scales were calculated from monthly precipitation for 1971-2016 from the NASA POWER database from four stations: Kalahandi, Kandhamal, Kesinga, and Nuapada. Precipitation from the preceding 1-12 months was used as input to account for persistence. We constructed three models: a Convolutional Neural Network (CNN), a Long Short-Term Memory (LSTM) model and a hybrid CNN-LSTM model. The models were compared based on correlation coefficient, RMSE, MAE, and NSE. The CNN-LSTM hybrid model was shown to outperform the CNN and LSTM models, with lower RMSE and higher NSE. This hybrid approach successfully leveraged CNN's spatial feature learning and LSTM's temporal feature learning, showcasing the potential for drought prediction and early warning in the Tel River Basin

KEYWORDS Drought forecasting; Standardized Precipitation Index; Deep learning; CNN; LSTM; Tel Basin.

1 INTRODUCTION

Drought is one of the most common and worst natural disasters and this adversely affects livelihood, agriculture, and water resources [1]. It is among the most difficult climatic extremities to predict since it starts slowly and persists over a long period and its complexity in altering space and time [2]. In India, where a large segment of the population depends on rain-fed farming [3], timely and accurate drought predictions are required to manage the risks effectively, prepare against disasters, and design policies that will safeguard the population. In this respect, Tel River Basin, which is a tributary of Mahanadi River located in the eastern side of the country, is particularly vulnerable to drought caused by weather [4]. The climate of the region is susceptible to incidences of droughts as it is subjugated by the monsoon and high rainfall variation between different seasons. Some of the districts within the basin that have been known to have suffered frequent droughts in the past are Kalahandi, Kandhamal, Kesinga, and Nuapada have had adverse effect on agricultural failure and socioeconomic losses [5].

It is also required that strong indices that can quantify abnormalities in rainfall in various temporal scales are in place in order to have an accurate forecast of drought [6]. Due to its simplicity, flexibility, and ability to show the level of drought at varying accumulation times, Standardized
Precipitation Index (SPI) has become one of the most used meteorological drought indicators [7]. There are various informations that SPI provides about various kinds of drought depending on the time scale chosen. Unlike longer windows such as SPI-6, SPI-9 and SPI-12, which displays seasonal to annual variations and thus is more useful in assessing the conditions of hydrological drought, short windows such as SPI-3 captures short-term changes in the moisture content that is relevant to agricultural drought [8]. By involving several SPI time scales simultaneously, researchers have a better chance of recording the persistence and development of drought conditions [9].

Machine learning (ML) and deep learning (DL) methods have a potential in the past few years to be used in hydro-meteorological forecasting activities [10]. Traditional statistical and soft computing models have been trying to predict drought namely autoregressive integrated moving average (ARIMA), genetic programming (GP) and artificial neural networks (ANN), but their success is often hampered in nonlinear, non-stationary climatic processes [11]. Conversely, nonlinear and hierarchical data patterns can be detected automatically by means of DL architectures. In particular, a special type of recurring neural networks known as the Long Short-Term Memory (LSTM) networks is more suitable to detect longterm temporal relationships in climate time series [12]. Similarly, they are capable of getting localized time-based features and performing dimensionality reduction using convolution and pooling. Convolutional Neural Networks (CNNs) that were initially designed to process different images and signals have been transformed to the time-series forecasting [13].

Although multi-scale temporal dynamics of drought indices have their distinct benefits, they might not be represented by an LSTM model or CNN model alone. Hybrid CNN and LSTM architectures have been suggested in other related domains such as speech recognition, energy forecasting, and flood prediction. In each of these models, the features are extracted with the help of CNN layers, and long-term dependencies are kept with the help of LSTM layers [14]. Nonetheless, CNN-LSTM hybrids are not yet actively used in the drought prediction in Indian settings [15], particularly whereby several SPI scales are taken into account across different stations.

In this paper, the authors investigate the effectiveness of CNN-LSTM hybrid architecture in the weather-based drought prediction in the Tel River basin of Odisha[14, 16]. The data on which the analysis is conducted is the monthly precipitation of four weather monitoring stations, namely, Kalahandi, Kandhamal, Kesinga, and Nuapada, between the years 1971 and 2016. By using this data, we calculated the SPI of 2 time-scales (SPI-3, SPI-6) to indicate a medium-term drought environment [17]. The forecasting experiments involved three models, which include CNN, LSTM, and the proposed CNN-LSTM hybrid. The model performance was determined by computing R-error, R-Square error, Root Mean Square error (RMSE), Mean Absolute error (MAE) and Nash Sutcliffe Efficiency (NSE) which when added together provide a comprehensive measure of the goodness of fit and the error level [18].

The objectives of this study include conducting an extensive analysis of the predictive capabilities of three popular deep learning models namely CNN, LSTM, and CNN-LSTM hybrid to predict the SPI of the four drought prone weather stations in the study area at different time scales, thereby analyzing their predictability in the diverse weather conditions of the region. Finally, the study will be used to demystify what advantages and disadvantages hybrid and single-architecture deep learning methods have in terms of their capacity to produce the multiscale and nonlinear variability of drought conditions. It is hoped that the gained knowledge in this case will assist in designing more effective earlywarning systems of drought in the Tel River basin and other monsoon-driven areas.

2 RESEARCH REGION AND DATA GATHERING

At present, the study is only confined to four weather stations in the Tel Basin of the state of Odisha, India. The four stations include Kalahandi, Kandhamal, Kesinga, and Nuapada; these are the stations which are affected by drought regularly. The Tel River basin is situated between latitudes 190°15′N North and 20°55′ N North and longitudes 82°03′ E East and 84°17′ E East covering an area of approximately 20, 490 square kilometers (see Fig. 1). Climate type is mostly that of tropical monsoon type which is characterized by mild winters, strong rainy season (June to September), and hot summers. The rainfall averages are between 12-50 mm/ year and over 75% of all rain is received during the southwest monsoon. Nevertheless, rainfall is highly variable over time and space, and during times of inadequacy or inconsistency of monsoon rainfall, meteorological droughts become widespread, particularly in western Odisha [19]. Different physiographic conditions are covered by the four stations: the Kalahandi and Kandhamal cover hilly and plateau districts, as well as the Kesinga and Nuapada cover transitional agroclimatic zones [20]. Due to such differences, the region is an ideal location to examine the dynamics of drought across various time scales [21]. The agricultural economy of these regions is very sensitive to the performance of monsoon and frequent droughts have been reported to have a great impact on crop failures, migration and economic distress. To perform the current analysis, monthly precipitation data were retrieved during 1971-2016 in the NASA-POWER data portal (https://power.larc.nasa.gov/data-accessviewer/). In the study area, figure 1 shows the positions of the four meteorological stations.

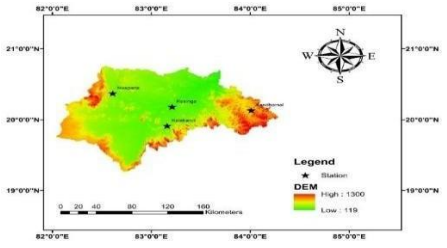

Figure 1: Study area, the Tel River basin, Digital Elevation model (DEM) showing the stations

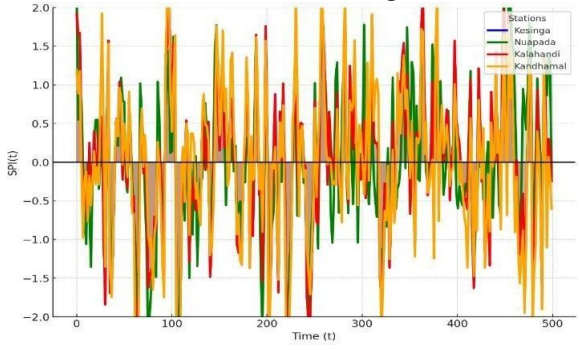

Figure 2: Plot showing the SPI-3 variation over time series for all stations

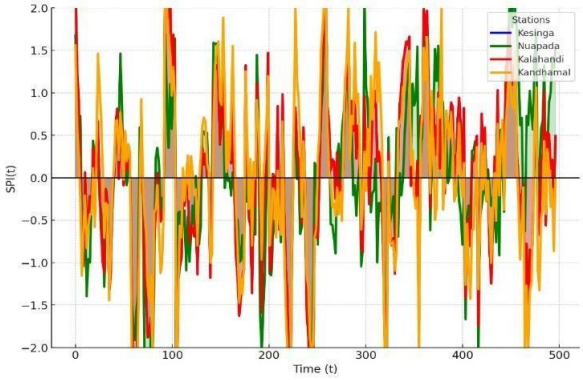

Figure 3: Plot showing the SPI-6 variation over the time series for all stations

3 METHODOLOGY 3.1The Standardized Precipitation Index (SPI)

SPI exclusively relies on precipitation and gives a normalized representation of anomaly at varying accumulation time intervals. It is obtained by summing up the amount of precipitation within the specified timeframe, applying an appropriate probability distribution to the summed-up data and transforming the cumulative probability into a standard normal deviate [22]. In the study, SPI was calculated at the two-time scales 3 & 6 months, hence accounting both the short-term agricultural moisture shortages and the long-term hydrological droughts [23].

The combined series of monthly precipitation were modeled to a Gamma distribution that is traditionally used to model SPI because of its merits. The cumulative distribution function was then used on every aggregated precipitation to get the probability. This probability was transformed by the inverse of the standard normal distribution to give the SPI value which is expressed in units of standard deviation relative to the historical record. The SPI values obtained in this process provide an unchanging index in all the stations and time lags. In this case, when the number is negative it refers to dry anomalies and when it is positive it refers to wet anomalies [24]. Values of SPI less than -1.0, -1.5 and -2.0, respectively, represent moderate, severe and extreme drought, within the classification thresholds applied to interpretative purposes (Table 1) [7]. Next, it was followed by the final series of SPI of each station and time scale as a target variable of drought prediction.

Table 1 Drought categories according to SPI classification

Range of SPI Values	Drought/Wetness Category
≥ 2.00	Extremely Wet

1.50 to 1.99	Very Wet
1.00 to 1.49	Moderately Wet
–0.99 to +0.99	Near Normal
–1.00 to –1.49	Moderately Dry
–1.50 to –1.99	Severely Dry
≤ –2.00	Extremely Dry

3.2 Long Short-Term Memory (LSTM)

A specific type of recurrent neural network (RNN) called Long Short-Term Memory (LSTM) networks was first [12] to deal with the problem of vanishing gradient related to the traditional RNNs. The simple RNNs are unable to store data when the sequences of information are long, but LSTM contains a memory cell structure, which enables the selective storage of information and forgetting information on every time step [25]. It is this property that enables them to be very useful in modelling the time dynamics of hydrometeorological indices like the Standardized Precipitation Index (SPI) [26].

The three gating mechanisms that govern each of the individual LSTM cells include the forget gate, the input gate, and the output gate. The flow of the information is regulated by these three gates, which also control the cell's state. Mathematically, an LSTM cell may be described as shown below: Mathematically, an LSTM cell can be expressed as follows:

$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f)$	(1)
$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i)$	(2)
$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$	(3)
$Ct = f \cdot Ct-1 + it \quad Ct$	(4)
$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$	(5)
$h_t = o_t \odot tanh(C_t)$	(6)

Here, f_t, i_t and o_t represent the input, output and forget gates respectively (Figure 4). While $C_{(t-1)}$ and C_t represent the memory states from the previous and current time steps, the hidden state from the previous time step is represented as $h_{(t-1)}$. The weight matrices are represented by W_f, W_i, W_c and W_o, and the bias terms are represented with the use of b_f, b_i, b_c, and b_o . The σ denotes the sigmoid activation function, and tan is the hyperbolic tangent function. The multiplication operation is an element-wise product of the matrices denoted by $\odot$. The traditional RNNs are based on a structure that is single-layered; on the contrary, LSTM networks possess more than one layer, and the interaction between these layers is necessary to increase memory retention and make the information processing more efficient. The initial part of an LSTM cell assesses the likelihood of retaining or forgetting the stored information that exists in the memory cell. As an example, the related pronouns used in a text prediction task should also be changed in case the gender of a subject is altered. This specific decision-making role occurs in the forget gate; a sigmoidal function is employed in a bid to control the flow of the information. The f_t value is determined using $h_{(t-1)}$ and X_t. The value will be zero to one. When f=1, the cell will store or store the memory of the last state output ($C_{(t-1)}$) but when the value is 0, the memory will be lost. New batch of data is then analyzed to determine what new data should be retained on the memory. This is done by the input gate and this has a sigmoid activation function and a hyperbolic tangent function to scan and control what is stored in C_t. Finally, the output gate decides what values will be passed to the next time step. And finally, in the last cell, i.e, the output cell, h_t, is computed using the sigmoid function, which selects parts of the memory cell, which are relevant to the output.

Various drought prediction works have recently used models using LSTM. Each of these models employs several LSTM layers and the quantity of input data directly influences the selection of the number of layers. However, excessive layers might cause a network to be more complicated, which may result in issues such as overfitting.

Figure 5 shows the structure of LSTM. The product of the LSTM layers is processed by a layer called the flatten and this is required to convert neurons into vectors. The flat data is then entered into the dense layers so as to establish the final output of the network. This is a recurrent neural network model that employs a recursive sequence method to determine the parameters and the weight matrices. These parameters are normally obtained by applying an optimization criterion like the Adam function. [27].

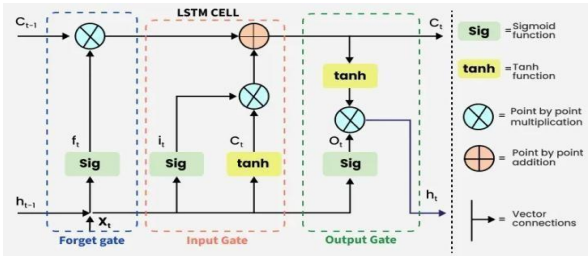

Figure 4: Illustrates the structure of a unit of LSTM

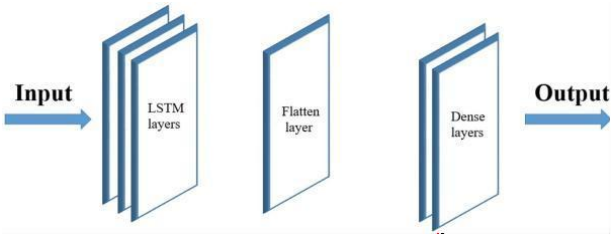

Figure 5:Dense layers, a flatten layer, and LSTM layers make up the LSTM-based model structure.

3.3 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a type of deep learning model that is used in both error correction and training of their input data using convolution operations. The CNNs are constructed to have several trainable layers just like other neural networks such as multilayer perceptron. Another significant advantage of CNNs is that it is able to automatically extract pertinent features of the input data. They are particularly helpful when there is a certain kind of correlation among the input because they lower the complexity of computing, and they help to decrease the problem of overfitting.

Figure 6 indicates a simple form of CNN structure. There are three major categories of layers in this structure, including convolutional, pooling, and fully linked layers. Convolutional, or filtering, layers are the main components of CNNs, and consist of a number of filters. The process of convolution of these filters and the input data results in the creation of a feature map. Max, or rather average, pooling layers or pooling layers, take out the most important values and in effect, decrease the dimensions of the data but still maintain important information. This action makes the processing more efficient and offers an opportunity to summarize the extracted features. The last layer of a CNN is the fully-linked layer which is structured similarly to the layers in a standard multilayer perceptron model. The outstanding advantage of CNNs compared to other forms of neural networks is that CNNs can automatically identify and extract features, without any human intervention. Hence, CNNs can be applied in regions where the input data of the data depend on each other. Also, these networks can be used to process large amounts of data and can reduce the chances of overfitting (Figure 7).

CNNs are a kind of deep learning model that can process images, but are also being applied to timeseries prediction since it is capable of automatically finding significant local characteristics [31]. In prediction of drought, CNNs are able to learn shortterm time variations and spikes in the SPI series with the assistance of convolutional filters on the lagged input sequences [32].

In a one-dimensional convolutional layer, a kernel (or filter), which has k elements, is applied to the input sequence to produce a few feature maps. The mathematical expression of the convolution operation of one of the feature maps may be expressed in the following form:

$$y_j(t) = \sigma\left(\sum_{i=1}^{n}\sum_{s=0}^{k-1} w_{i,j}(s)\cdot x_i(t-s) + b_j\right) \quad (7)$$

Here, $x_i(t)$ is the input sequence, $w_{i,j}$ represents the filter weights connecting input channel i to output channel j k, ,is the kernel size, b_j is the bias, and σ() is the activation function (commonly ReLU). Pooling layers (e.g., max pooling or average pooling) are used after convolution for dimensionality reduction and retaining the most important features [33]. The outputs of these convolutional and pooling layers are then flattened and forwarded to the dense layers, which then map the extracted features to the target SPI values.

In this research, benchmark models include employing a CNN. The input consists of lagged SPI sequences at various timescales, which then enables the CNN to learn patterns of local temporality. Examples of local temporality include abrupt changes in rain or short episodes of drought. Although CNN has a limitation in capturing longterm temporality, it is powerful in detecting and summarizing local variations impacting short-term drought.

All the gating operations let the network keep relevant temporal information through the longhorizon and omit the irrelevant (or noise) information. A lot of the choices made in the design the input format, kernel size, filter numbers, units of CNN, and pooling window size result in tradeoffs between training/future and the efficiency of the model and the accuracy of the predictions [34]. The architecture for this study is shown in Figure 6 and was chosen after the empirical tuning with the training dataset. The model parameters were tuned to reduce the mean squared error between the observed output SPI y_t and prediction $\hat{y}_t$:

$$L = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2, \quad (8)$$

using the Adam optimizer and hyperparameters (learning rate, batch size, number of epochs, early stopping patience, regularization).

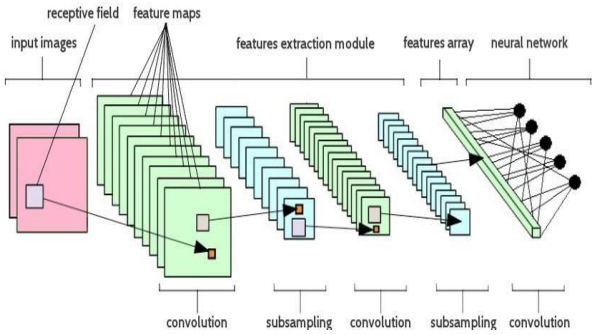

Figure 6: CNN structure consists of convolutional layers, pooling, and fully connected layers

3.4 The Hybrid of CNN-LSTM Model

Combining convolutional and recurrent structures has been typical for a number of sequence-modelling tasks [35]. In this work, for the CNN–LSTM architecture, the pipeline first seizes local temporal features through one-dimensional convolutional filters and then sends the acquired features to LSTM layers that remember and carry long-term dependencies; this mapping is indicated as:

$Z = Conv1D(X; W_c, b_c)$	(9)
$(h_t, C_t) = LSTM(Z_t, h_{t-1}, C_{t-1})$	(10)
$y_t = W_d h_t + b_d,$	(11)

Where $X = \{x_1,....,x_t\}$ is the input sequence of lagged SPI values, $Z = \{Z_1,.....,Z_t\}$ denotes the CNN-extracted feature sequence, (h C_t, $_t$) are the LSTM hidden and cell states at time t, and $^{y}{}_{t}$is the expected SPI.

The upper part of the hybrid model has a convolutional component that detects important local structures and streamlines the redundancy of the raw sequence. One 1-D convolutional feature map can be expressed as:

$$y_j(t) = \sigma(\sum_{i=1}^{n} w_{i,j}(s) x_i(t-s) + b_j), \quad (12)$$

where $^{X}{}_{i}$ (.) denotes the ith input channel (for univariate SPI-based inputs n=1), $^{w}{}_{i\,j,}$ are filter weights, k is the kernel size, $^{b}{}_{j}$ is a bias term, and σ( ) is the activation (e.g., ReLU). Optionally, a pooling operator such as max-pooling may be applied to $^{y}{}_{j}$ in order to produce a lowerdimensional summary $^{p}{}_{j}$:

$$p_j(t) = \max_{u \in [0,m-1]} y_j(t-u), \quad (13)$$

that assists in stabilizing local appearance reactions and minimizes the amounts of parameters that is introduced to the recurrent block.

The LSTM block, which follows the convolutional stack, is a sequence-based learning block that models its internal gates and the memory cell and its internal dynamics are described by the conventional gating equations.

4 MODEL DEVELOPMENT

Three & Six months Standardized Precipitation Index (SPI-3, SPI-6) were both modelled and forecasted with one month lead-time using the values of the past months as input parameters. The input characteristics were built based on lagged SPI values of the past months, which allowed the models to identify the temporal relationship of the drought index series.

The forecasting problems can be expressed as:

$$SPI_t^{(k)} = f(SPI_{t-1}^{(k)}, SPI_{t-2}^{(k)}, \ldots, SPI_{t-m}^{(k)}) \quad (14)$$

with t representing the time step under consideration, m the number of lagged months to be treated as predictors, and k the time scale of accumulation of the SPI (e.g. 3 or 6 months). Each station (Kalahandi, Kandhamal, Kesinga, and Nuapada) is covered by the whole dataset (552 months, 1971 to 2016). To develop the model, the dataset was randomly split into 80 percent training and 20 percent testing samples and no particular year was assigned to any of the groups. Three models were applied in this study in order to compare the predictive performance.

Monthly values were ordered chronologically in all SPI time series which corresponded to a station and accumulation scale. To simulate forecasting conditions in the real world, this temporal partitioning ensured that only past data was used to train the models, and more recent unseen data was used to evaluate the models. The 80:20 strategy provided a moderate model of assessing the generalization abilities of the CNN, LSTM and CNN-LSTM models through a combination of a powerful training dataset and a reliable hold-out test dataset.

The choice of the optimization method is an important procedure in the training of deep learning models as it defines how the model parameters are updated through the back-propagation [36]. The Adaptive Moment Estimation (Adam) optimizer was used to train all three model architectures of this paper, including CNN, LSTM, and CNN-LSTM. Adam uses momentum in the form of exponentially decaying averages of past gradients and obtains the advantages of two prior algorithms, AdaGrad and RMSProp, to become one of the most popular deep learning optimizers [37]. This enables Adam to deal with sparse gradients, non-stationary targets, and noisy data more effectively than conventional stochastic gradient descent (SGD) does [38]. Adam was selected because of both theoretical and real-life considerations. By definition, drought time series such as SPI are nonlinear, noisy and have long-term relationships; therefore, it would be optimal to have an optimizer that can dynamically adjust learning rates and accelerate convergence without having to do much manual fiddling. In practice, Adam tends to work well with default settings of a learning rate of 0.001, 0.9 as the beta 1 value, and 0.999 as the beta 2 value. These settings were utilized in the present work. Adam

optimizer was chosen as the training algorithm, and the implementation was done in Python with the usage of the Keras deep learning framework. Adam repeatedly obtained stable convergence across all models in testing and never exhibited the oscillating behavior characteristic of the traditional SGD. Its performance in the modeling framework as the default optimizer was explained by its high performance. When training deep learning models, the number of epochs or complete cycles of the model over the training data has to be specified. The number of epochs is very important because not enough training may result in underfitting, and excessive training may lead to the phenomenon of overfitting when the model begins to memorize patterns in training at the cost of generalizing [39]. The number of epochs was first put to a relatively high upper limit in this study (e.g., 200) in order to give the models time to converge. Nevertheless, instead of using a fixed number of epochs, an early stopping criterion was added as the most effective way of deciding the optimal training duration in real time. Early termination is another common regularization method used in deep learning, and this method tracks the performance of the model on a validation set. In particular, after every epoch, the loss in the validation was monitored. In case the validation loss was not improving after a specified number of consecutive epochs (also referred to as the patience parameter, which was 20 in this case) the training was prematurely stopped, and the model weights with the best validation loss were reloaded. This ensured that training was terminated at the stage of optimal generalization and it helped in avoiding underfitting and overfitting. Early stopping combined with an upper limit on the number of epochs was an efficient use of computational power since models could easily stop early in the middle of the 200 epochs. Rather, the training commonly converged after 70120 epochs, depending on the architecture and SPI timescale, with strong and unbiased model testing.

5 MODEL PERFORMANCE EVALUATION

This paper measures the performance of the models based on different statistical measures, amongst which are the absolute error measures, including Root Mean square error (RMSE) and Mean Absolute error (MAE), and goodness-of-fit measurements, such as Nash-Sutcliffe Efficiency (NSE) [40].

These metrics have mathematical formulas as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(SPI_{ci} - SPI_{pi})^2}{n}} \quad (15)$$

$$MAE = \frac{\sum_{i=1}^{n}| SPI_{ci} - SPI_{pi} |}{n} \quad (16)$$

$$NSE = 1 - \left[\frac{\sum_{i=1}^{n}(SPI_{ci} - SPI_{pi})^2}{\sum_{i=1}^{n}(SPI_{ci} - \overline{SPI_c})^2} \right] \quad (17)$$

Here, n refers to the overall number of data points to be utilized in the process of training and testing. The terms $^{SPI}{}_{ci}$ and $^{SPI}{}_{pi}$ are the Standardized correspond to the observed and predicted values of the Standardized Precipitation Index (SPI), while

$\overline{\quad}$

$\overline{\quad}$

SPI_c and SPI_p signify their respective mean values.

6 RESULT AND DISCUSSION

6.1 Prediction of SPI-3 at all the stations

In the case of Kandhamal, the three models are similar in short lags, whereas CNN–LSTM is obviously better in long lags (Figure 7). At lag-11, the hybrid attains higher explained variance and lower errors than both baselines (CNN–LSTM: R^2 = 0.594, RMSE = 0.686, MAE = 0.525; LSTM: R^2 = 0.210, RMSE = 0.829; CNN: R^2 = 0.274, RMSE = 0.795). A similar advantage holds at lag-12 (CNN–LSTM: R^2 = 0.545, RMSE = 0.726; LSTM: R^2 = 0.228, RMSE = 0.805; CNN: R^2 = 0.208, RMSE = 0.816) (Table 2), indicating that the hybrid captures year-scale structure in SPI3 more effectively. At Kalahandi, overall skill is similar across models, but the pure CNN exhibits instability at certain lags (e.g., at lag-1 CNN: R^2 = 0.055, RMSE = 0.965; at lag-7 CNN: $R^2 \approx 0.001$, RMSE = 0.918). In comparison, CNN-LSTM is more stable and generally comparable to LSTM at higher lags (e.g., at lag-5 CNN-LSTM R^2 = 0.302, RMSE = 0.784 vs. LSTM R^2 = 0.303, RMSE = 0.783), thus the hybrid is the less risky operational model. For Kesinga, LSTM peaks at lag-2 (LSTM: R^2 = 0.579, RMSE = 0.677), but the CNN–LSTM takes the lead at lag-5 (CNN-LSTM: R^2 = 0.534, RMSE = 0.701; LSTM R^2 = 0.295, RMSE = 0.788) and again near lag-11 (CNNLSTM R^2 = 0.265, RMSE = 0.773; LSTM R^2 = 0.231, RMSE = 0.790). Elsewhere, the hybrid is much like LSTM, with the advantage and no harm to stability.

At Nuapada, the most predictable site, the CNNLSTM is strongest at lags 10–12, consistently outperforming LSTM (CNN-LSTM lag-10: R^2 = 0.54, RMSE = 0.66 vs. LSTM: R^2 =0.32, RMSE = 0.77; CNNLSTM lag-11: R^2=0.51, RMSE = 0.68 vs. LSTM: R^2 =0.40, RMSE = 0.75; CNN-LSTM lag-12: R^2 =0.42, RMSE = 0.75 vs. LSTM: R^2= 0.30, RMSE = 0.78). The hybrid has the high and constant ability on other lags as that of LSTM. In general, CNN-LSTM is the most

suitable model to forecast SPI-3 since it does not exhibit the apparent failures of pure CNN (Kalahandi lag-1 and lag-7), and has the best results with key lags (Kandhamal (11 and 12), Kesinga (5 and 11) and Nuapada (10 and 12). It is also approximately equal to LSTM at other lags (e.g. Kalahandi lag-5).

Figure 8 represents Taylor diagrams that sum up the model performance of SPI-3 at each of four locations. The CNN-LSTM model has lower centered RMS differences and a higher correlation coefficient, when compared to CNN and LSTM.

Based on the quantitative measures (R^2 = 0.594, RMSE = 0.686), the hybrid model fits the observed reference point the most, which proves it to be more consistent with the real series of SPI-3. The argument in favor of the balanced bias-variance trade-off of CNN-LSTM is that all three models work equally well at Kalahandi, and CNN-LSTM is a bit more stable. CNN-LSTM has an improved correlation with Kesinga again where the optimal fit is at the middle standard deviations but pure CNN has a slight deviation which can be interpreted as underfitting at the particular standard deviations. This is supported by the Taylor diagram, which reveals that CNN-LSTM has the lowest RMS error and the highest correlation at Nuapada, which is in line with its superiority in Table 2 and the numerical statistics. On the whole, CNN-LSTM hybrid provides the most trustworthy and realistic reconstruction of monitored SPI-3 variations in each station, particularly in complicated time conditions, which are observed by the Taylor plots. Figure 9 demonstrates the distribution of actual and expected SPI-3 values of each of the stations. The distribution of all stations (red) and projected distribution (green) have almost the same value, which shows that CNN-LSTM model effectively recovers the amplitude and time fluctuations of SPI-3, The observed and projected SPI-3 median of Kandhamal are almost identical and this proves that the CNNLSTM model is able to reconstruct the centraltendency of the model. Although the variability of the projected set is a bit less at Kalahandi, the overall distribution stays the same, implying that there should be no overfitting and generalization is steady. The model is a precise representation of the bimodal behavior of Kesinga which shows the established trend of dryness and wetness. In the case of Nuapada, the reduced forecasted range demonstrates that the extremes are slightly underestimated, but the mean is the same, which proves that CNN-LSTM does not lose the SPI structure even at the most predictable points.

Taken together, the quantitative metrics (R^2, RMSE, MAE) as reported above are graphically supported by the violin-box plots, which indicate that the CNN-LSTM model is able to reproduce the statistical distribution of SPI-3 in the varying hydroclimatic contexts.

Table 2. Test-set performance for medium-term drought prediction (SPI-3) at four Tel River Basin stations across lags 1–12; best values are shown in bold.

		LSTM					CNN					CNN-LSTM				
	LAG	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
Kandhamal	**1**	0.529	0.231	0.790	0.638	0.231	0.632	0.381	0.764	0.600	0.381	0.470	0.059	1.000	0.780	0.059
	2	0.522	0.247	0.846	0.684	0.247	0.527	0.260	0.838	0.682	0.260	0.513	0.235	0.853	0.687	0.235
	3	0.367	0.050	0.947	0.769	0.050	0.468	0.143	0.899	0.734	0.143	0.431	0.098	0.923	0.748	0.098
	4	0.505	0.216	0.858	0.695	0.216	0.516	0.215	0.859	0.694	0.215	0.499	0.186	0.874	0.700	0.186
	5	0.555	0.278	0.820	0.678	0.278	0.573	0.303	0.805	0.658	0.303	0.588	0.321	0.795	0.646	0.321
	6	0.554	0.257	0.812	0.674	0.257	0.590	0.327	0.773	0.630	0.327	0.520	0.220	0.832	0.665	0.220
	7	0.545	0.243	0.827	0.668	0.243	0.583	0.313	0.787	0.638	0.313	0.559	0.275	0.809	0.664	0.275
	8	0.558	0.266	0.817	0.663	0.266	0.582	0.314	0.790	0.630	0.314	0.516	0.204	0.851	0.696	0.204
	9	0.533	0.236	0.835	0.678	0.236	0.562	0.285	0.807	0.653	0.285	0.481	0.159	0.875	0.705	0.159
	10	0.542	0.293	0.823	0.670	0.240	0.559	0.264	0.810	0.655	0.264	0.500	0.192	0.848	0.685	0.192
	11	0.531	0.210	0.829	0.683	0.210	0.563	0.274	0.795	0.648	0.274	**0.772**	**0.594**	**0.686**	**0.525**	**0.594**
	12	0.540	0.228	0.805	0.647	0.228	0.521	0.208	0.816	0.661	0.208	0.739	0.545	0.726	0.561	0.545
Kalahandi	**LAG**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
	1	0.523	0.222	0.795	0.635	0.222	0.461	0.055	0.965	0.749	0.055	0.460	0.071	0.971	0.754	0.071
	2	0.530	0.235	0.822	0.673	0.235	0.510	0.242	0.819	0.665	0.242	0.500	0.226	0.827	0.675	0.226
	3	0.279	0.000	0.943	0.767	0.000	0.463	0.156	0.866	0.700	0.156	0.366	0.071	0.909	0.735	0.071
	4	0.467	0.191	0.847	0.687	0.191	0.615	0.371	0.770	0.581	0.371	0.476	0.176	0.855	0.688	0.176
	5	0.566	0.303	0.783	0.631	0.303	0.575	0.313	0.778	0.632	0.313	**0.576**	**0.302**	**0.784**	**0.640**	**0.302**
	6	0.553	0.273	0.776	0.631	0.273	0.562	0.282	0.772	0.615	0.282	0.494	0.176	0.827	0.666	0.176
	7	0.530	0.240	0.801	0.648	0.240	0.278	0.001	0.918	0.735	0.001	0.507	0.220	0.811	0.654	0.220

	8	0.559	0.274	0.780	0.629	0.274	0.653	0.420	0.705	0.549	0.420	0.505	0.209	0.815	0.657	0.209
	9	0.560	0.280	0.776	0.627	0.280	0.574	0.304	0.763	0.622	0.304	0.529	0.245	0.795	0.653	0.245
	10	0.573	0.297	0.764	0.613	0.297	0.554	0.264	0.781	0.637	0.264	0.518	0.225	0.801	0.657	0.225
	11	0.523	0.222	0.795	0.635	0.222	0.635	0.392	0.733	0.561	0.392	0.498	0.198	0.807	0.659	0.198
	12	0.585	0.313	0.740	0.608	0.313	0.519	0.170	0.813	0.663	0.170	0.291	0.059	0.866	0.705	0.059
Kesinga	**LAG**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
	1	0.572	0.309	0.780	0.640	0.309	0.460	0.068	0.969	0.753	0.068	0.460	0.076	0.973	0.756	0.076
	2	0.761	0.579	0.677	0.529	0.579	0.510	0.240	0.819	0.664	0.240	0.499	0.227	0.826	0.674	0.227
	3	0.291	0.085	0.941	0.766	0.003	0.462	0.160	0.864	0.698	0.160	0.357	0.073	0.908	0.740	0.073
	4	0.499	0.215	0.835	0.675	0.215	0.510	0.222	0.831	0.659	0.222	0.479	0.171	0.858	0.693	0.171
	5	0.562	0.295	0.788	0.639	0.295	0.578	0.320	0.774	0.628	0.320	**0.732**	**0.534**	**0.701**	**0.545**	**0.534**
	6	0.533	0.244	0.792	0.643	0.244	0.558	0.267	0.780	0.623	0.267	0.493	0.194	0.818	0.652	0.194
	7	0.517	0.220	0.811	0.655	0.220	0.561	0.282	0.778	0.631	0.282	0.533	0.250	0.796	0.639	0.250
	8	0.547	0.266	0.785	0.627	0.266	0.577	0.305	0.764	0.620	0.305	0.516	0.225	0.807	0.650	0.225
	9	0.567	0.292	0.769	0.625	0.292	0.548	0.262	0.785	0.644	0.262	0.498	0.209	0.813	0.663	0.209
	10	0.566	0.292	0.768	0.607	0.288	0.558	0.272	0.777	0.631	0.272	0.513	0.228	0.800	0.668	0.228
	11	0.529	0.231	0.790	0.638	0.231	0.534	0.213	0.799	0.650	0.213	0.548	0.265	0.773	0.621	0.265
	12	0.556	0.264	0.766	0.617	0.264	0.141	0.496	0.827	0.674	0.141	0.312	0.041	0.874	0.697	0.041
Nuapada	**LAG**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
	1	0.634	0.382	0.747	0.576	0.382	0.570	0.162	0.872	0.665	0.162	0.565	0.157	0.875	0.668	0.157
	2	0.646	0.415	0.725	0.593	0.415	0.645	0.414	0.726	0.590	0.414	0.646	0.415	0.725	0.593	0.415
	3	0.557	0.305	0.797	0.643	0.305	0.592	0.336	0.780	0.633	0.336	0.603	0.359	0.766	0.618	0.359
	4	0.458	0.179	0.853	0.692	0.179	0.602	0.380	0.758	0.602	0.340	0.493	0.200	0.843	0.676	0.200
	5	0.641	0.397	0.745	0.582	0.397	0.648	0.408	0.739	0.573	0.341	0.631	0.390	0.750	0.585	0.390
	6	0.634	0.382	0.747	0.576	0.382	0.625	0.381	0.747	0.567	0.381	0.616	0.353	0.764	0.583	0.353
	7	0.623	0.368	0.762	0.597	0.368	0.634	0.383	0.753	0.575	0.383	0.648	0.402	0.741	0.571	0.402
	8	0.594	0.322	0.788	0.622	0.322	0.484	0.186	0.864	0.663	0.186	0.627	0.387	0.753	0.568	0.387
	9	0.631	0.387	0.753	0.596	0.387	0.640	0.397	0.747	0.577	0.397	0.602	0.343	0.776	0.604	0.343
	10	0.620	0.318	0.765	0.602	0.372	0.618	0.370	0.767	0.577	0.370	0.734	0.539	0.660	0.507	0.539
	11	0.643	0.401	0.747	0.561	0.401	0.614	0.347	0.780	0.588	0.347	**0.719**	**0.514**	**0.680**	**0.530**	**0.514**
	12	0.569	0.303	0.780	0.613	0.303	0.556	0.266	0.801	0.610	0.266	0.649	0.418	0.748	0.596	0.418

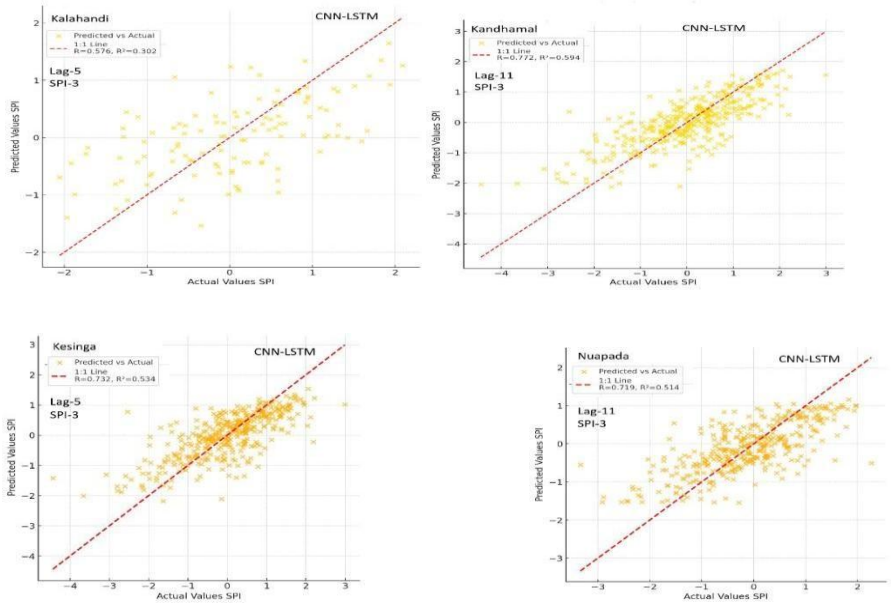

Figure 7: Plots showing the best model at all the stations for SPI-3 Prediction

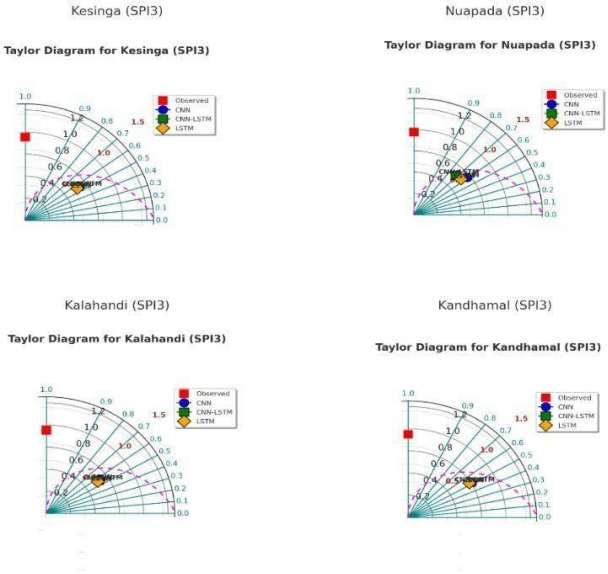

Figure 8: Taylor diagrams for SPI-3 across four stations (Kandhamal, Kalahandi, Kesinga, and Nuapada) comparing the performance of CNN, LSTM, and CNN–LSTM models.

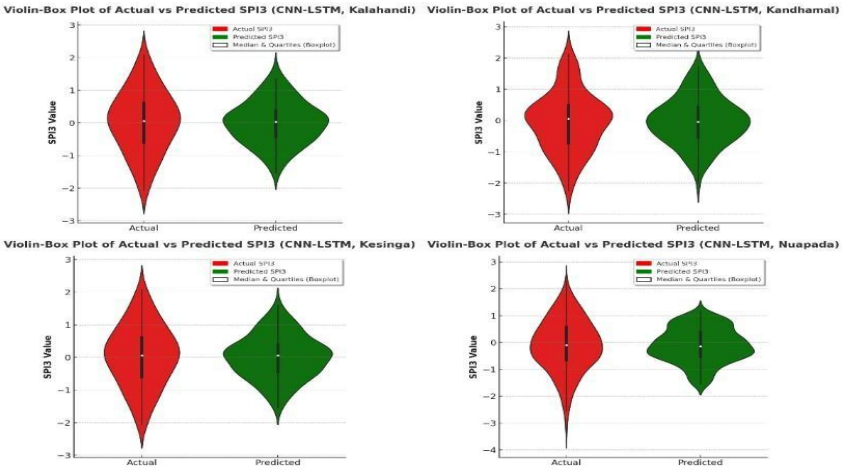

Figure 9: ***Violinplots of actual versus predicted SPI-3 values for four stations (Kalahandi, Kandhamal, Kesinga, and Nuapada) using the CNN–LSTM model.***

6.2 Prediction of SPI-6 at all the stations

The hybrid CNN-LSTM tends to equal or exceed practically useful short-to-moderate lags (Figure 10). In the case of Kandhamal, all models have the lowest errors at a lag of approximately 2 months; the CNN-LSTM model has the minimum error (RMSE=0.62) and the highest explanatory power (R^2=0.46) slightly better than the LSTM, and thus, it is appropriate to state that convolutional feature extraction and subsequent recurrent integration is better able to capture the short-range persistence. The lag-2 performance of the hybrid is significant even though the LSTM is competitive at a number of other lags in this station since it matches the range of memory that is characteristic of multi-month drought anomalies.

The overall difference between architectures is very small at Kalahandi, and the CNN-LSTM has a regular the two separate LSTM and CNN at all four of the stations, the most noticeable benefits being at tendency to avoid CNN drawbacks and equal or surpass LSTM at a number of lags.

Specifically, the hybrid is consistent in skill at lags 8 to 12 and better in errors and has higher R^2 than the LSTM at lag-11 where the recurrent branch seems to smoothen the convolutional features and convert small seasonal-scale features with incremental improvements. Notably, the CNN exhibits a significant degradation at lag-7, which is not the case of the CNN-LSTM, highlighting the strength of the hybrid model in the event of the shift of the informative temporal scale.

In the case of Kesinga, skill grows as soon as some minimum temporal context is availed. Since lag-2, CNN-LSTM is very correlated and efficient as compared to the LSTM, but slightly lowering errors at

certain lags (e.g., 6 and 8 and 1 and 2). The fact that the hybrid and the recurrent baseline are nearly equivalent here implies that once the information about the context between months is present, the two models are able to extract similar information but the small but repeated advantages of the hybrid at the selective lags are an indication that localized convolutional filters could be used to augment recurrent memory in transferring sub-seasonal variations to better SPI-6 predictions.

Nuapada is the most foreseeable location, and it is the place where the edge of the hybrid is most significant at one, operationally significant lag. The CNN-LSTM achieves almost perfect fidelity (R= 1.00, very small error) at lag-3, which is significantly higher than the LSTM (R= 0.95) as indicated in Table 3. Although this extraordinary performance necessitates standard protection against target leakage and inappropriate scaling, its coincidence with high level of short-to-moderate predictability in this station is also in line with a sharp three-month antecedent signal in SPI-6. In addition to lag-3, the hybrid is generally comparable to the LSTM at lag 2 and lag 5 and 6 with low errors and high efficiencies respectively of the overall predictability of the site. The combination of the information of each station indicates that the CNN-LSTM is the most effective operational model in the prediction of SPI-6. Its performances are the most dramatic at short lags, especially lag-2 in Kandhamal and lag-3 in Nuapada, where the LSTM layer integrates local temporal structures represented by convolutional filters across a multi-month memory. In the case where the hybrid fails to dominate, it is at least just as stable as the LSTM and it never experiences isolated failures that the pure CNN does. These results validate the mechanical explanation of drought indices by the convolutional-recurrent pipeline which includes cumulative hydroclimatic memory and short-range anomaly structure.

Figure 11 uses Taylor diagrams that simultaneously represent correlation, centered RMS error, and standard deviation that are normalized to summarize model performance of SPI-6. The CNNLSTM cluster is nearest to the reference point of all stations which implies that its correlation and error properties are equal in relation to the series observed. Hybrid model of Kandhamal is the one that results in the least difference in RMS, and maximum correlation; this is as expected as there is a numerical advantage near lag-2. The models all work well in Kalahandi but CNN-LSTM has a slightly better correlation and more stable variance representation, which proves that it is resilient to longer lags. As per the conclusion of the text that the two designs work in a similar way when sufficient time context is provided, CNN-LSTM and LSTM markers by Kesinga are almost similar in terms of their behavior. The CNN-LSTM at Nuapada nearly fits the observed reference which indicates nearperfect correlation and minimal error which is in line with the high performance (R 1.0) in Table 3.3. All these diagrams together give graphical evidence that the CNN-LSTM effectively reproduces the amplitude and phase of the observed SPI-6 variability and thus it is effective in recreating the short- to moderate-term drought persistence and still has better or equivalent skill in all the stations.

Figure 12 presents the distributional correspondence between the observed and predicted values of the CNN-CNN-LSTM on the values of the SPI-6 of each individual station. The displayed projected and actual distributions of Kandhamal are almost similar, which proves that the hybrid was able to reflect the central tendency and the general distribution of SPI-6 correctly, as it is consistent with its better lag-2 performance. The fact that the model stabilizes at larger lags would also be supported by the observation that the variance is well represented by the model at Kalahandi with only a minor compression of extremes. The CNN -LSTM of Kesinga recreates the median and interquartile range of the SPI-6 and proves the ability of the model to preserve subseasonal variations provided that enough temporal context is provided. At Nuapada, the fact that the predicted and actual distributions are very close also serves as evidence of the incredible fidelity of the model ($R \approx 1.00$ at lag-3) as witnessed in the quantitative analysis. All in all, the Violin plots support the conclusion that the CNN-LSTM does not only offer high predictive accuracy but also preserves the natural structure of variability of SPI6, which suggests that the learned representations of the CNN-LSTM are effective in capturing the probabilistic properties of the drought dynamics at the regional scale. The fact that the peaks and troughs of all the stations are temporally consistent indicates that there is a high level of regionalization of drought occurrence, which could be as a result of the monsoonal rainfall distribution patterns on western Odisha. The fact that the CNN-LSTM reproduces these oscillations in other stations has shown that it can replicate the temporal changes in terms of evolution and persistence of the SPI-6, and thus confirming its structural appropriateness in the modelling of the sub-seasonal drought patterns.

Table 3. Test-set performance for medium-term drought prediction (SPI-6) at four Tel River Basin stations across lags 1–12; best values are shown in bold.

		LSTM					CNN					CNN-LSTM				
	LAG	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
	1	0.655	0.438	0.604	0.459	0.438	0.511	0.225	0.829	0.654	0.225	0.625	0.266	0.714	0.534	0.266

Kandhamal	**2**	0.684	0.446	0.624	0.473	0.446	0.527	0.260	0.838	0.682	0.260	**0.691**	**0.459**	**0.617**	**0.469**	**0.459**
	3	0.631	0.370	0.670	0.510	0.370	0.468	0.143	0.899	0.734	0.143	0.626	0.366	0.672	0.510	0.366
	4	0.677	0.447	0.630	0.481	0.447	0.516	0.215	0.859	0.694	0.215	0.658	0.412	0.650	0.498	0.412
	5	0.645	0.401	0.660	0.509	0.401	0.573	0.303	0.805	0.658	0.330	0.648	0.391	0.666	0.516	0.391
	6	0.630	0.376	0.679	0.512	0.376	0.590	0.327	0.773	0.630	0.327	0.621	0.344	0.696	0.533	0.344
	7	0.586	0.313	0.718	0.547	0.313	0.583	0.313	0.787	0.638	0.313	0.629	0.361	0.692	0.534	0.361
	8	0.623	0.367	0.694	0.518	0.367	0.582	0.314	0.790	0.630	0.314	0.648	0.380	0.639	0.509	0.380
	9	0.601	0.348	0.699	0.514	0.348	0.562	0.285	0.807	0.653	0.285	0.619	0.354	0.695	0.545	0.354
	10	0.637	0.375	0.682	0.525	0.375	0.559	0.264	0.810	0.655	0.264	0.620	0.348	0.697	0.536	0.348
	11	0.664	0.421	0.657	0.497	0.421	0.563	0.274	0.795	0.648	0.274	0.649	0.396	0.671	0.516	0.396
	12	0.665	0.407	0.667	0.512	0.407	0.520	0.208	0.816	0.661	0.208	0.582	0.319	0.715	0.559	0.319
Kalahandi	**LAG**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
	1	0.644	0.377	0.669	0.512	0.377	0.583	0.182	0.735	0.555	0.183	0.582	0.173	0.736	0.553	0.173
	2	0.624	0.359	0.653	0.502	0.359	0.622	0.349	0.658	0.506	0.349	0.620	0.355	0.655	0.506	0.355
	3	0.571	0.288	0.693	0.525	0.288	0.580	0.287	0.694	0.534	0.287	0.552	0.261	0.706	0.536	0.261
	4	0.626	0.382	0.645	0.495	0.382	0.639	0.387	0.642	0.494	0.387	0.618	0.347	0.663	0.505	0.347
	5	0.581	0.315	0.683	0.536	0.315	0.574	0.267	0.710	0.535	0.267	0.587	0.308	0.687	0.527	0.308
	6	0.539	0.239	0.721	0.554	0.239	0.590	0.298	0.693	0.533	0.298	0.582	0.292	0.696	0.530	0.292
	7	0.593	0.313	0.690	0.520	0.313	0.381	0.040	0.816	0.627	0.040	0.586	0.291	0.701	0.534	0.291
	8	0.590	0.288	0.709	0.544	0.288	0.604	0.294	0.706	0.539	0.294	0.612	0.358	0.673	0.512	0.358
	9	0.555	0.275	0.715	0.549	0.275	0.633	0.362	0.671	0.509	0.362	0.598	0.333	0.686	0.528	0.333
	10	0.607	0.335	0.685	0.515	0.334	0.620	0.324	0.691	0.525	0.324	0.580	0.299	0.703	0.537	0.299
	11	0.570	0.306	0.707	0.530	0.306	0.612	0.353	0.682	0.530	0.353	**0.637**	**0.368**	**0.674**	**0.515**	**0.368**
	12	0.593	0.329	0.232	0.329	0.529	0.633	0.353	0.686	0.519	0.353	0.602	0.357	0.684	0.520	0.357
Kesinga	**LAG**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
	1	0.601	0.327	0.679	0.522	0.327	0.460	0.056	0.964	0.748	0.056	0.460	0.079	0.975	0.758	0.079
	2	0.623	0.354	0.656	0.503	0.354	0.626	0.361	0.652	0.502	0.361	0.624	0.356	0.655	0.504	0.356
	3	0.575	0.300	0.687	0.520	0.300	0.573	0.277	0.698	0.539	0.277	0.551	0.260	0.707	0.536	0.260
	4	0.626	0.382	0.645	0.495	0.382	0.632	0.375	0.648	0.502	0.375	0.620	0.347	0.662	0.506	0.347
	5	0.578	0.316	0.683	0.541	0.316	0.610	0.339	0.671	0.520	0.339	0.597	0.315	0.684	0.529	0.315
	6	0.520	0.224	0.729	0.570	0.224	0.601	0.327	0.679	0.522	0.327	0.581	0.284	0.700	0.537	0.284
	7	0.565	0.289	0.702	0.541	0.288	0.595	0.295	0.699	0.537	0.295	0.587	0.295	0.699	0.535	0.295
	8	0.603	0.317	0.694	0.536	0.317	0.594	0.271	0.717	0.545	0.271	0.590	0.293	0.706	0.550	0.293
	9	0.586	0.336	0.684	0.521	0.336	0.629	0.355	0.674	0.519	0.355	0.584	0.304	0.701	0.538	0.304
	10	0.603	0.320	0.693	0.534	0.320	0.606	0.298	0.704	0.535	0.298	0.586	0.298	0.704	0.535	0.298
	11	**0.629**	**0.381**	**0.667**	**0.514**	**0.381**	0.644	0.377	0.669	0.512	0.377	0.629	0.373	0.671	0.513	0.373
	12	0.579	0.326	0.700	0.535	0.326	0.629	0.342	0.692	0.524	0.342	0.641	0.384	0.670	0.520	0.245
Nuapada	**LAG**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**	**R**	**R^2**	**RMSE**	**MAE**	**NSE**
	1	0.701	0.526	0.523	0.408	0.526	0.740	0.488	0.638	0.477	0.488	0.728	0.476	0.645	0.479	0.476

	2	0.772	0.593	0.565	0.414	0.593	0.775	0.595	0.564	0.409	0.595	0.742	0.548	0.596	0.437	0.548
	3	0.946	0.888	0.297	0.240	0.888	0.600	0.522	0.622	0.475	0.522	**0.998**	**0.996**	**0.034**	**0.025**	**0.999**
	4	0.628	0.387	0.642	0.493	0.387	0.621	0.362	0.655	0.514	0.362	0.617	0.342	0.665	0.514	0.342
	5	0.737	0.536	0.615	0.466	0.536	0.767	0.583	0.583	0.425	0.583	0.740	0.536	0.615	0.457	0.536
	6	0.728	0.517	0.629	0.482	0.517	0.743	0.536	0.617	0.462	0.536	0.731	0.516	0.630	0.481	0.516
	7	0.739	0.545	0.726	0.561	0.545	0.739	0.525	0.627	0.472	0.525	0.723	0.500	0.644	0.498	0.500
	8	0.707	0.468	0.669	0.510	0.468	0.674	0.404	0.708	0.543	0.404	0.732	0.518	0.637	0.486	0.518
	9	0.702	0.474	0.669	0.503	0.474	0.744	0.529	0.633	0.493	0.529	0.738	0.523	0.637	0.487	0.523
	10	0.716	0.498	0.652	0.494	0.498	0.763	0.569	0.603	0.463	0.569	0.745	0.540	0.623	0.476	0.540
	11	0.715	0.497	0.648	0.482	0.495	0.757	0.557	0.607	0.464	0.557	0.734	0.527	0.628	0.493	0.527
		0.689	0.465	0.664	0.497	0.465	0.760	0.564	0.600	0.466	0.564	0.616	0.327	0.745	0.551	0.327

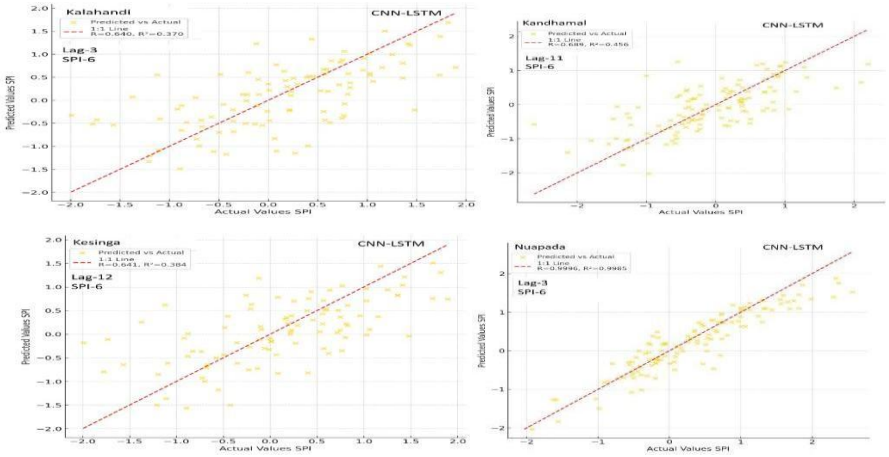

Figure 10: Plots showing the best model at all the stations for SPI-6 Prediction

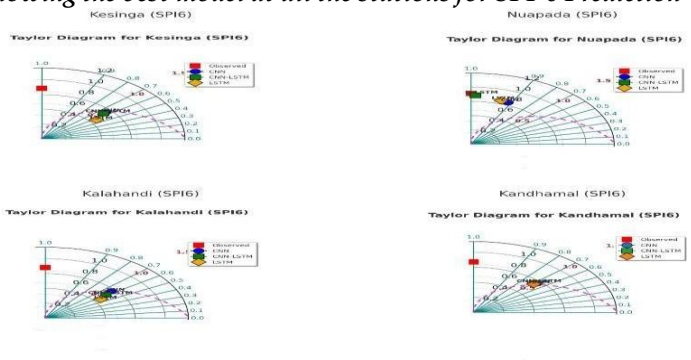

Figure 11: Taylor diagrams of SPI-6 at four stations (Kandhamal, Kalahandi, Kesinga, and Nuapada) of CNN, LSTM, and CNN-LSTM model performance.

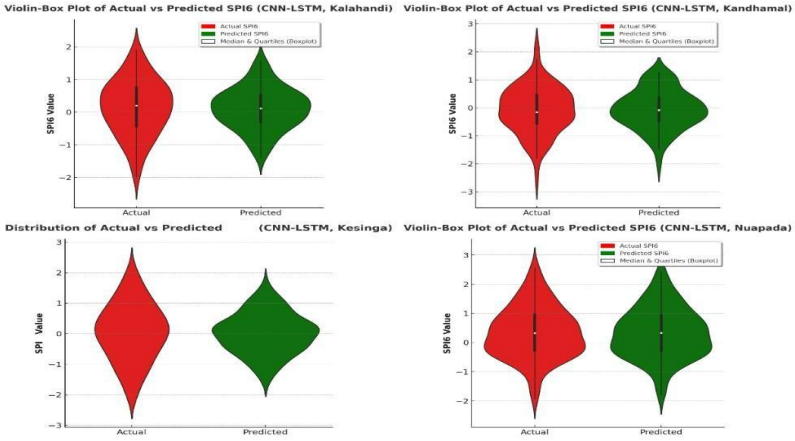

Figure 12: Violin plots of the actual and predicted SPI-6 values of four stations (Kandhamal, Kalahandi,

Kesinga and Nuapada) with the CNN-LSTM model

7 CONCLUSION

This paper compares medium-term drought forecasting over the Tel River Basin in terms of using SPI-3 and SPI-6 at four sites (Kandhamal, Kalahandi, Kesinga, and Nuapada) and three deep-learning models: LSTM, CNN, and a hybrid CNN-LSTM. The CNN-LSTM was the most reliable in general, provided the best and the most consistent results in Kandhamal (lags 11-12), Kesinga (lag 5 and approximately 11) and Nuapada (lag 10-12), but was generally similar to the baseline models (LSTM and CNN). Performance is highest at lags 2 to 4 with further gains at lags 10 to 12. The hybrid strength is best realized in the case where the pure CNN is found to be unstable, and therefore, integrating convolutional feature extraction together with the LSTM result in a stable representation of the SPI signal. The future research that encompasses other hydro-climatic predictors, quantification of uncertainty, and broader cross-basin experiments can further justify the concept of generalizability, but the available data makes it evident that CNNLSTM is a good starting point in this basin and issued scenario.

REFERENCES

Bordi, I. and A. Sutera, Drought variability and its climatic implications. Global and Planetary Change, 2004. 40(1-2): p. 115-127.

Seager, R. and M. Ting, Decadal drought variability over North America: Mechanisms and predictability. Current Climate Change Reports, 2017. 3(2): p. 141-149.

Anshuka, A., F.F. van Ogtrop, and R. Willem Vervoort, Drought forecasting through statistical models using standardised precipitation index: a systematic review and meta-regression analysis. Natural Hazards, 2019. 97(2): p. 955-977.

Panda, A., Vulnerability to climate variability and drought among small and marginal farmers: a case study in Odisha, India. Climate and Development, 2017. 9(7): p. 605-617.

Sahoo, B.B., Critical appraisal of different drought indices of drought predection & their application in kbk districts of odisha. 2014.

Zhu, Y., et al., Combined use of meteorological drought indices at multi-time scales for improving hydrological drought detection. Science of the Total Environment, 2016. 571: p. 1058-1068.

McKee, T.B., N.J. Doesken, and J. Kleist. The relationship of drought frequency and duration to time scales. in Proceedings of the 8th Conference on Applied Climatology. 1993. California.

Prajapati, V., et al., Evaluation of time scale of meteorological, hydrological and agricultural drought indices. Natural hazards, 2021. 109(1): p. 89-109.

Merabti, A., et al., Spatial and time variability of drought based on SPI and RDI with various time scales. Water resources management, 2018. 32(3): p. 1087-1100.

Kadam, C.M., U.V. Bhosle, and R.S. Holambe, Deep learning-driven regional drought assessment: an optimized perspective. Earth science informatics, 2024. 17(2): p. 1523-1537.

Alquraish, M., et al., SPI-based hybrid hidden Markov–GA, ARIMA–GA, and ARIMA–GA–ANN models for meteorological drought forecasting. Sustainability, 2021. 13(22): p. 12576.

Hochreiter, S. and J. Schmidhuber, Long short-term memory. Neural computation, 1997. 9(8): p. 1735-1780.

Dehghani, A., et al., Comparative evaluation of LSTM, CNN, and ConvLSTM for hourly short-term streamflow forecasting using deep learning approaches. Ecological Informatics, 2023. 75: p. 102119.

Barzegar, R., M.T. Aalami, and J. Adamowski, Short-term water quality variable prediction using a hybrid CNN-LSTM deep learning model. Stochastic Environmental Research and Risk Assessment, 2020. 34(2): p. 415-433.

Elbeltagi, A., et al., Drought indicator analysis and forecasting using data driven models: case study in Jaisalmer, India. Stochastic Environmental Research and Risk Assessment, 2023. 37(1): p. 113-131.

Mishra, S.S. and R. Nagarajan, Spatio-temporal drought assessment in Tel river basin using Standardized Precipitation Index (SPI) and GIS. Geomatics, Natural Hazards and Risk, 2011. 2(1): p. 79-93.

Umran Komuscu, A., Using the SPI to analyze spatial and temporal patterns of drought in Turkey. 1999.

Gyaneshwar, A., et al., A contemporary review on deep learning models for drought prediction. Sustainability, 2023. 15(7): p. 6160.

Prabhakar, A., et al., Assessment of regional-level long-term gridded rainfall variability over the Odisha State of India. Applied Water Science, 2019. 9(4): p. 93.

ROUT, R., et al., Bt-Cotton Cultivation in Western undulating Agro-climatic zone of Odisha: Farmers' Perceptions, Economic Viability, and Challenges-A Case Study.

Ma, L., et al., Propagation dynamics and causes of hydrological drought in response to meteorological drought at seasonal timescales. Hydrology Research, 2022. 53(1): p. 193-205.

Guttman, N.B., Accepting the standardized precipitation index: a calculation algorithm 1. JAWRA Journal of the American Water Resources Association, 1999. 35(2): p. 311-322.
Lloyd-Hughes, B. and M.A. Saunders, A drought climatology for Europe. International journal of climatology, 2002. 22(13): p. 1571-1592.
Hänsel, S., A. Schucknecht, and J. Matschullat, The Modified Rainfall Anomaly Index (mRAI)—is this an alternative to the Standardised Precipitation Index (SPI) in evaluating future extreme precipitation characteristics? Theoretical and applied climatology, 2016. 123(3): p. 827-844.
Graves, A., Long short-term memory. Supervised sequence labelling with recurrent neural networks, 2012: p. 37-45.
Reusser, D., et al., Analysing the temporal dynamics of model performance for hydrological models. Hydrology and earth system sciences, 2009. 13(7): p. 999-1018.
Zhang, Z. Improved adam optimizer for deep neural networks. in 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS). 2018. Ieee.
Liu, T., et al., Implementation of training convolutional neural networks. arXiv preprint arXiv:1506.01195, 2015.
Rudenko, O., O. Bezsonov, and O. Romanyk, Neural network time series prediction based on multilayer perceptron. Управління розвитком= Development Management, 2019. 18(1): p. 23.
Banerjee, I., et al., Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. Artificial intelligence in medicine, 2019. 97: p. 79-88.
Sadouk, L., CNN Approaches for Time Series. Time Series Analysis: Data, Methods, and Applications, 2019. 57.
Tian, W., et al., Drought prediction based on feature-based transfer learning and time series imaging. IEEE Access, 2021. 9: p. 101454-101468.
Van Der Maaten, L., E.O. Postma, and H.J. Van Den Herik, Dimensionality reduction: A comparative review. Journal of machine learning research, 2009. 10(66-71): p. 13.
Alzubaidi, L., et al., Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 2021. 8(1): p. 53.
Danandeh Mehr, A., et al., A novel intelligent deep learning predictive model for meteorological drought forecasting. Journal of Ambient Intelligence and Humanized Computing, 2023. 14(8): p. 10441-10455.
Wythoff, B.J., Backpropagation neural networks: a tutorial. Chemometrics and Intelligent Laboratory Systems, 1993. 18(2): p. 115-155.
Heredia, C., Modeling AdaGrad, RMSProp, and Adam with Integro-Differential Equations. arXiv preprint arXiv:2411.09734, 2024.
Zhang, J., et al., Why ADAM beats SGD for attention models. 2019.
Aliferis, C. and G. Simon, Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI. Artificial intelligence and machine learning in health care and medical sciences: Best practices and pitfalls, 2024: p. 477-524.
Newaz, M., Comparing the performance of time series models for forecasting exchange rate. 2008.