International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 2, April 2025, pp. 1868~1879
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i2.pp1868-1879

Towards an automated weather forecasting and classification using deep learning, fully convolutional network, and long short-term memory

Nilesh Shelke[1], Sudhanshu Maurya[1], Rupali Ithape[2], Zarina Shaikh[3], Rachna Somkunwar[3], Amit Pimpalkar[4]
[1]Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India
[2]Department of Electronics and Communication, School of Engineering and Science (SOES), MIT-ADT University, Pune, India
[3]Department of Computer Engineering, Dr. D. Y. Patil Institute of Technology-Pimpri, Pune, India
[4]School of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Ramdeobaba University, Nagpur, India

Article Info

Article history:

Received Mar 27, 2024
Revised Oct 24, 2024
Accepted Oct 31, 2024

Keywords:

Deep learning
Fully convolutional network
Long short-term memory
VGG16
Weather forecasting

ABSTRACT

Historically, weather forecasting was unreliable and imprecise, relying on intuition and local knowledge. Inaccurate weather forecasts can cause severe impacts on agriculture, construction, and daily life. Existing methods struggle with rural and urban weather prediction, requiring faster and more accurate solutions. This research proposes a deep learning system using real-time images to address this challenge. This research employs a deep learning model fully convolutional network-long short-term memory (FCN-LSTM) to analyze images and predict weather conditions. In this case, the model forecasts a sunny and cloudy environment, which facilitates defining the ideal conditions for every given climatic zone in the weather classification model. The model is trained on a dataset of weather images obtained from Kaggle. The performance of the proposed model FCN-LSTM achieves an accuracy of 96.88% and a validation accuracy of 91.22%. Also, the mean squared error (MSE) is 7.11, which is significantly lower and supports efficient enhancement in weather forecasting. This significant improvement demonstrates the potential of deep learning for real-time weather forecasting. The model provides efficient weather classification, enabling informed decision-making across various sectors. This research lays the foundation for automated weather analysis using deep learning, eliminating human bias and improving accuracy.

Corresponding Author:

Sudhanshu Maurya
Department of Computer Science and Engineering, Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India
Email: dr.sm0302@gmail.com

1. INTRODUCTION

Since weather is the foundation of all practical systems, including those related to agriculture, energy generation, tourism, transportation, navigation, and climate, it is an essential field of scientific study [1], [2]. Precise classification averts fatalities, reduces the impact of severe weather disasters, and generates significant financial revenue, aiding emergency management [3]. Early in the 20th century, scientists solved non-linear differential equations by hand, laying the groundwork for current weather forecasting. Today's Supercomputers can accurately foresee up to months in advance by solving these equations for millions of points every time step. The goal of this task, which meteorologists refer to as weather forecasting and classification, is to anticipate a location's weather based on a meteorological component, such as temperature or wind speed. Nonetheless, weather forecasting models are becoming increasingly complicated, necessitating a constant increase in processing power. These models can take hours, days, or even weeks to produce findings, which restricts their capacity to make useful forecasts quickly.

One of the most significant uses of artificial intelligence and machine learning is weather forecasting, which allows one to predict the atmospheric conditions for a given place at a given time. People have attempted to make informal weather classification for millennia, but with the advent of sophisticated tools and vast data sets, the task has become considerably more straightforward. Meteorology and quantitative data collected for a particular place are used to construct weather classification, showing the anticipated atmospheric conditions change. A century ago, weather forecasting was an erratic, unpredictable, and untrustworthy process. The data collection observations were erratic and intermittent. The forecaster used intuition-based estimates, familiarity with local climatology, and imprecise and dispersed extrapolation techniques; theoretical physics principles were either wholly absent or hardly mentioned in actual forecasting. It appeared that forecasting was more of an art than a science. The inaccuracies in weather forecasting can be attributed to several factors, including i) There is a significant discrepancy between the forecasted and actual time; ii) Intricate mathematical and statistical calculations that demand significant processing power; and iii) Mistakes made during measurement recording and insufficient knowledge of meteorology.

Weather forecasting provides a multitude of advantages, including safeguarding lives and property by issuing early warnings for extreme weather events, supporting informed decision-making in agriculture, optimizing transportation and logistics planning, aiding in energy management, fostering economic growth through cost-effective investments, promoting public health preparedness, facilitating educational and recreational planning, and supporting scientific research and development in various fields such as climate change and weather modification. These benefits underscore the critical role of weather forecasting in enhancing safety, economic prosperity, and overall quality of life. Aside from that, several innovative ideas and methods have been introduced due to scientists' and researchers' repeated attempts to solve various issues to achieve weather forecast accuracy. However, the level of accuracy attained using these strategies fell short of expectations. A place's weather refers to the state of the atmospheric conditions, including wind speed, temperature, precipitation, humidity, cloud cover, and other atmospheric phenomena, in a specific location for a brief period. The weather is a natural occurrence crucial to global atmospheric balance. Several factors can cause the atmospheric state to fluctuate, and occasionally, it can get severe enough to jeopardize property and human life. The term "severe weather" describes this. If the underlying causes of these disorders are identified, future harm can be prevented, and preventative measures can be taken. Computer vision researchers have discussed virtual worlds for three decades, yet there are certain limitations because of insufficient data. Weather intelligence systems are facing issues in predicting accurate weather conditions in rural and urban areas of the country. Because the previous traditional examination took a long time, scientists have turned to artificial intelligence and deep learning techniques to analyze and predict weather conditions.

This research aims to create a system that uses real-time photographs to forecast the weather in a particular area. This technology contributes to the agricultural sector, construction sector, and everyday human life routine decisions. Deep learning enables precise and faster and accurate weather classification in different areas. Governments increasingly utilize deep learning for various applications, including fraud detection, customer relationship management, computer vision, vocal artificial intelligence, and natural language processing. Deep learning, mainly through multi-layered artificial neural networks (ANNs), has demonstrated high accuracy in detection, identification, and classification tasks. Research has shown the effectiveness of deep learning models, such as convolutional neural network (CNN) and recurrent neural network (RNN), in areas like classification, prediction, and public sector applications, including cyber defense, traffic congestion prediction, and disease spread tracking. CNN and RNN are utilized for weather analysis and image or video detection [4]–[6]. The increasing inflation due to shortage and loss of farming, which sees unpredictable weather incidents as its main enemy, and unwanted rains and poor classification of weather conditions are the driving forces behind the proposed initiative. Using digital images and surveillance footage, the researchers in this study created a deep-learning fully convolutional network-long short-term memory (FCN-LSTM) model to predict weather conditions.

The research aims at solving the problem of using real time images to forecast the weather since manual methods have proven to lack the necessary precision. The work shows a detailed method of predicting the weather through the utilization of deep learning methods. The following are the methodologies highlighted in the sharing section:

a. This approach is a combination of the models and aims at increasing the accuracy of measure and efficiency. The inputs of the weather images are preliminary processed by VGG16 model to extract its features before classification by the FCN and LSTM layers.

b. This layer involves feature extraction from the images, and the accurate predictions of the features. An LSTM layer is employed for classifying the results on the weather conditions according to the features identified by the FCN.
c. This layer appears especially helpful in capturing the long-term dependencies which may be present in the data and in making predictions for the sequential data. The study illustrates the use of both FCN and LSTM layers to guarantee an efficient weather prediction.
d. The FCN-LSTM has a classification accuracy of 96.88% in the weather classification, this figure is higher than the one obtained from other methods by more than 5%.
e. Since the proposed model is for automation and scalable, Kaggle's graphics processing unit (GPU) kernels have been provided for training and analysis. This makes it possible to make large scale predictions while also giving real time results making it practical for use in weather prediction.

This work is divided into six sections, each reviewing the state-of-the-art and conventional methods for weather classification and offering a new methodological recommendation based on the research. Section 2 presents the pertinent work. The outlining technique for LSTM and FCN is detailed in section 3. The suggested approach for classifying weather is described in section 4. In section 5, the results of the proposed weather classification approach are analyzed and evaluated. Section 6 concludes the research findings.

2. RELATED WORKS

Several recent literature reviews have been conducted on weather image classification using various deep-learning techniques for classifying weather images based on their visual characteristics. This summary highlights key findings and approaches presented in different research works. Li and Luo [7] proposes a novel weather image classification model, combining a vision transformer (VIT) for capturing global image relationships and a dual enhanced attention module to extract deeper semantic features, aiming for improved accuracy and addressing the limitations of traditional methods. Rani *et al.* [8] proposed an EfficientNet and dual attention block for weather image classification. These studies provide valuable insights into using deep learning techniques such as CNN, VIT, and EfficientNet for weather image classification. Mittal and Sangwan [9] proposed a framework applying transfer learning for weather image classification. They analyze the performance of diverse deep CNNs built with CNN, Keras, and TensorFlow, achieving high accuracy in classifying images into different weather categories. Naufal and Kusuma [10] focused on classifying weather states using deep learning. Their investigation involved a multi-class weather dataset and the application of deep learning models, specifically comparing four CNN architectures: MobileNetV2, VGG16, DenseNet201, and Xception. Xiao *et al.* [11] presented a weather-image classification model to address challenges in weather-image classification and achieve high-quality automatic classification. Their work emphasizes the potential of deep learning in meteorology and the effectiveness of deep CNNs for weather image classification tasks. Bai *et al.* [12] provided a broader overview of deep learning applications in meteorology. They discussed using deep learning for image analysis and recognition of meteorological data, including radar images and other elements.

Additionally, the researchers reviewed the current state of deep learning applications in weather forecasting worldwide. Those studies collectively demonstrate the power of deep learning in automatically classifying weather images with high accuracy. Different approaches utilizing various convolutional neural network architectures and novel techniques like attention modules have been explored, showcasing the potential for further advancements in weather prediction and analysis through deep learning methods. Unlike statistical models, these models were able to increase prediction accuracy. Nevertheless, the practical implementation was unfeasible due to the restricted forecast range of 30–180 minutes and the challenges encountered in reaching solution convergence. Traditional machine learning includes linear regression and support vector machines, which have been explored as forecasting candidates and are generally significantly less computationally demanding than neural networks. For example, researchers projected humidity and air temperature over 3 hours using a traditional machine learning XGBoost model composed of gradient-boosted decision trees. According to the data, the root temperature of mean squared error was 1.77 degrees Celsius. For weather classification, deep learning is recommended for many reasons, even though classical machine learning algorithms yield relatively decent results. Non-linearity, crucial for forecasting weather evolution, cannot be modelled by conventional algorithms. According to Hewage *et al.* [13], their machine-learning algorithms estimate weather conditions twelve hours in advance more accurately than traditional weather forecasting.

The use of support vector machines and their modifications for time series and non-linear data categorization, as well as short-term series forecasting, has been the focus of recent work [14]–[16]. Similarly, Shao *et al.* [17] noted that complicated wind forecasting is not a good fit for statistical or conventional machine learning approaches, and they attributed this requirement to the turbulent and chaotic nature of wind. Unlike machine learning, which can learn from smaller datasets, deep learning requires large amounts of data. An increasing number of domains, such as finance, sugarcane yield categorization, and power load forecasting, use deep learning networks for time series forecasting [18]. The accuracy of weather forecasting could be significantly enhanced by deep learning, and its applications could grow rapidly. Weyn *et al.* [19] ensemble modelling of distinct CNN models-each with a different set of weights and beginning conditions-improved the accuracy of weather classification. In their evaluation of a multi-layer perceptron, an LSTM model, and a hybrid LSTM or CNN model, Roy [20] concluded that models with more intricate architectures generally perform better, while according to Ravuri *et al.* [21], their neural network model performs better than other weather classification methods in 89% of cases when it comes to precipitation classification.

Precipitation forecasting has shown neural networks to be especially promising. It was demonstrated that a MetNet model created at Google could correctly forecast precipitation over eight hours. Several models, including CNNs and LSTMs, were employed at various phases in this hybrid strategy. Even with its strong performance, the model needs many data. Met-Net2, a deep-learning weather model, outperformed state-of-the-art models for up to 12-hour precipitation forecasting in the Continental United States [22]. After analyzing numerous neural network architectures, Fu *et al.* [23] combined a one-dimensional CNN with a bidirectional-long short-term memory (Bi-LSTM) [24]–[26] to predict wind speed, relative humidity, and ground air temperature over seven days. With over a million nodes, the final model was built using weather station data from 10 Beijing-area stations. Despite its scale and complexity, its quantitative performance compared to local weather measurements was dubious. Two recent developments are hybrid LSTM for cloud movement classification and LSTM and CNN [26] for drought prediction. Wind forecasting is crucial to estimating wind power and load, and deep learning has recently been used [27]. Most applications accelerated classification by up to 24 hours, focusing on the short term. According to current literature, large-scale predictions with LSTM dominant point forecasts and CNN-variant architectures demonstrate how quickly deep learning technologies are used in weather forecasting.

Recent research uses deep learning, particularly CNNs, and transfer learning to effectively classify diverse weather conditions from images, paving the way for improved weather prediction and analysis. They also highlight the use of various deep learning models, approaches, and the accuracies achieved in classifying weather images. Researchers have developed a new wind speed prediction model using graph attention networks (GAT) and Bi-LSTM layers [28]. This model uses historical weather data's spatial and temporal properties to predict wind speed more efficiently than previous algorithms. The model was tested using precise meteorological data from wind farms in Tetouan, Morocco, and outperformed other models in terms of MSE. The GAT-BiLSTM model's MSE was 0.1573, significantly lower than those of other models. Some limitations, such as the need for large datasets and high computing resources, are also discussed in these works. However, there are several research obstacles concerning short-term prediction. While most applications were created in wind farm settings with "simple" weather patterns, they are more unpredictable in urban areas due to increased signal turbulence. Additionally, forecasts begin to deteriorate after a few hours, and there is no perfect forecast length.

3. PROPOSED METHOD

The proposed methodology comprises two steps: weather analysis and identification of weather conditions using different pictures in the datasets. The picture dataset by Kaggle is used to validate the suggested FCN-LSTM model. As shown in Figure 1, a novel weather forecasting system takes local images as input, applies pre-processing and feature extraction through VGG16 and FCN, uses LSTM for prediction, and finally combines all features for accurate weather forecasting. The flow shows input pictures of the place or area passed to the model. The relevant feature extraction uses the VGG16 layers as part of the data preparation stage. The VGG16 model is used to analyze these images to obtain the necessary image features to extract and analyze meteorological information. At this stage, the photos are strongly associated with features that need to be collected and moved to the FCN layer, which is situated above the LSTM layer and under the VGG16. The FCN, which sits above the LSTM recurrent layer, receives the pre-processed images. The FCN sends the input to the LSTM for classification after doing the most accurate analysis on it using the features it has gathered. The LSTM forecasts the data to ascertain the future weather conditions after getting input from the FCN layer. The meteorological dataset is used to train and evaluate the proposed model. The model achieves sequential classification using the FCN and LSTM approaches, where LSTM is used for further classification after the initial few predictions. The FCN acts as both a feature extractor and a predictor, ensuring accurate weather forecasting.

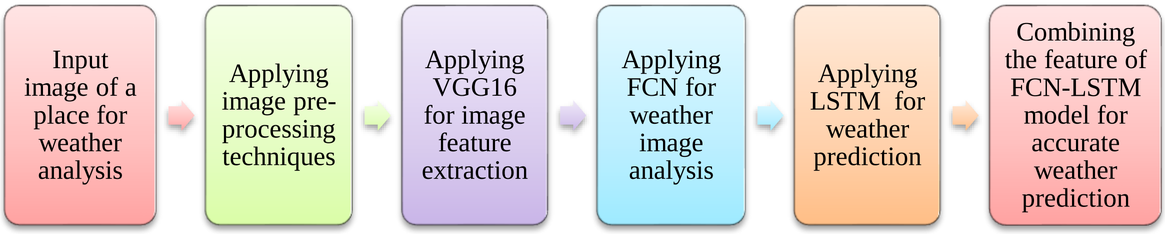

Figure 1. Proposed innovative methodology for weather forecasting

3.1. Feature extraction using VGG16

In this study, we use the power of CNNs to extract meaningful features from weather images. Specifically, we employed a pre-trained VGG16 model, which has been trained on a vast dataset of images and has learned to identify various features such as edges, lines, and shapes. By discarding the final classification layers, we were able to use VGG16 as a feature extractor, allowing it to extract low-level features like edges and lines as well as more complex features like shapes and textures. To ensure the model can efficiently understand the visual data, we pre-processed the images by converting them to PNG format and resizing them to a uniform size of 50×50 pixels. This method allows VGG16 to consistently interpret the information within each image and extract relevant features.

The VGG16 model is composed of four blocks of convolutional layers, each playing a crucial role in identifying specific weather conditions. The first block contains two convolutional layers with 64 filters each, followed by a max-pooling layer. Similarly, the second block contains two convolutional layers with 128 filters each, the third block contains three convolutional layers with 256 filters each, and the last block contains three convolutional layers with 512 filters each, followed by a max-pooling layer. The first block extracts low-level features; the second block extracts more complex features, and so on. The combination of these blocks enables the model to accurately classify weather images. Once the features are extracted, we use class-weighted loss functions to handle imbalanced datasets, where some classes have significantly more instances than others. By assigning higher weights to rare weather conditions, the model is encouraged to focus more on these conditions and reduce overfitting on frequently occurring weather conditions. The VGG16 model outputs a vector for each image, summarizing the extracted features. This vector becomes a new representation of input image data, capturing the essence of the image without the raw pixel details. These feature vectors are then passed to an FCN for image classification.

3.2. FCN for image classification

CNNs are used by FCNs to convert picture pixels into pixel classes. An FCN extracts the height and breadth of intermediate feature maps and helps recognize the images with pixel values close to those of images with similar weather conditions. Deep learning models predict a class for each pixel, creating a detailed, point-by-point match between the original image and its classification outcome. An FCN instance is established to provide output. It aids in correctly categorizing the climatic conditions, except for the fully linked and final global average pooling layers, which are closer to production. To the best of our understanding, the LeNet adaptation introduced the idea of expanding a convolutional network to arbitrary-sized inputs to recognize strings of digits. To successfully decode the output, researchers use Viterbi decoding because their network can only process one-dimensional input strings. This technique is necessary to effectively handle the output from their network, ensuring the intended outcomes are achieved. The CNN outputs are expanded into 2D maps of identification scores for additional blocks. Furthermore, all convolutional computation has been used in many-layered networks in the current era. Fully convolutional inference is used in semantic segmentation in picture reconstruction and sliding window identification.

The FCN is built upon the VGG16 model, which extracts features from input images. These features are then passed to the FCN, which consists of multiple convolutional and pooling layers. The FCN consists of four 2D convolutional of 32, 64, 128, and 256 filters, kernel size of 3×3, an activation function as rectified linear unit (ReLU), and four MaxPooling2D layers that work together to classify the images into different weather categories. The convolutional layers apply filters to detect specific patterns and features in the images, while the pooling layers downsample the feature maps to reduce spatial dimensions and increase the number of channels. The output of the pooling layers is flattened into a one-dimensional feature vector, which is then passed to fully connected layers designed to classify the images into different weather categories. The fully connected layers are followed by a SoftMax activation function, which produces a probability distribution over the different weather categories. The class with the highest probability is selected as the predicted class for the input image. Our FCN architecture is designed to effectively classify weather images by smearing the power of deep learning and CNN.

A CNN has the following dimensions for each data layer: $h \times w \times d$, where d is the feature or channel dimension, while w and h are the spatial dimensions. The picture is the first layer of color-channel $h \times w$ pixels. A continuous path connects reachable areas within an image, and higher layer placements correspond to these regions. The key elements of these algorithms are activation functions, pooling, and convolution-related geographical coordinates, which are all that are needed and are limited in the range of input regions they may operate on. These functions write x_{ij} for the data vector at position (i, j) in a specific layer and for the following layer to produce outputs y_{ij} using (1).

$$y_{ij} = f_{ks}\left(\{x_{si} + \delta_i, s_j + \delta_j\} \, 0 \leq \delta_i, \delta_i \leq k\right) \tag{1}$$

In layer specification, the activation function denotes non-linearity; Max pooling finds the maximum activation within a sliding window, while convolution uses matrix multiplication to slide a filter and learn features, and average pooling averages all activations within a sliding window. The symbols 'k' and 's' represent kernel size and stride (subsampling factor). The transformation rule, applicable to kernel size and stride, remains consistent throughout the composition, providing a standardized functional form for various layer types.

$$f_{ks} \circ g_{k's'} = (f \circ g)_{k'+(k-1)s', ss'} \tag{2}$$

A deep filter FCN is a neural network exclusively comprising non-linear layers that compute non-linear filters, contrasting with general deep networks that compute generic non-linear functions. The unique capability of an FCN is its automatic conversion of inputs of varying sizes into outputs with corresponding spatial dimensions, potentially requiring resampling. Tasks are defined by real-valued loss functions constructed using FCNs. When the loss function is spatially summed, the final layer gradient is the sum of individual gradients computed for each spatial feature map, effectively integrating information from all image regions, which is expressed as $\ell(x; \epsilon) = \sum ij\ell' (xij; \epsilon)$. Stochastic gradient descent (SGD) on the full image's ℓ is equivalent to SGD on ℓ' when each final layer receptive field is treated as a mini-batch. Efficient feedforward computation and backpropagation occur when computing layer-by-layer over the entire image rather than patch-by-patch. The efficient feedforward computation and backpropagation are particularly advantageous in cases of significant receptive field overlap.

3.3. LSTM for weather classification

Once the input layer's dimensions have been altered and prepared to use the LSTM for classification, it is sent via the LSTM cell. Tanh, or the hyperbolic tangent, is the LSTM cell's activation function. Additionally, the LSTM cell features a dropout rate to prevent data overfitting. These features of LSTM allow it to precisely recall the input image's form and long-term dependencies in a specific pattern. The LSTM-based model is designed to accurately classify weather images into their corresponding categories, which are very close to the actual weather conditions. The original LSTM model made significant advances; among them was creating channels where the gradient may run continually by employing self-loops. Making the self-loop variable's weight trainable would be superior to fixing it, offering greater flexibility and potentially improving performance. This self-loop weight can be gate-adjusted (by another hidden unit) to dynamically modify the integration time scale. We may conclude that even in the case of a parameter-fixed LSTM, the input sequence may change the time scale of integration as the model itself generates the time constants in this case. A broad range of applications, such as machine translation, photo captioning, speech recognition, handwriting creation, and unlimited handwriting recognition, have shown the LSTM to perform very well. LSTM block structure, as shown in Figure 2, consists of a cell state and three gates: a forget gate, an input gate, and an output gate. These gates regulate the flow of information into and out of the cell state, allowing LSTMs to selectively retain or discard information as it flows through the network. The following graphic displays the forward propagation equations needed for a deep recurrent network's construction.

Deep learning structures, particularly LSTM cells, have been shown to outperform simpler models in handling long-term dependencies. Unlike regular neurons, LSTM cells have an internal loop that controls information flow, allowing them to learn sequential data more effectively. This unique ability makes them particularly powerful for tasks such as machine translation and speech recognition, where sequential data processing is crucial. RNNs represent a category of artificial neural networks tailored explicitly for processing sequential data, encompassing language, speech, and time series information. Unlike conventional RNNs, which lack shared hidden units, these networks utilize recurrent connections between cells to enhance their sequential processing capabilities. Each artificial neuron unit calculates an input characteristic, contributing to the state if activated by the sigmoidal input gate. LSTMs use forget, input, and output gates to control information flow. The forget gate decides what to remember, the input gate selects new information, and the output gate determines what influences the final output. Each gate uses a sigmoid function except the input gate, which offers flexibility.

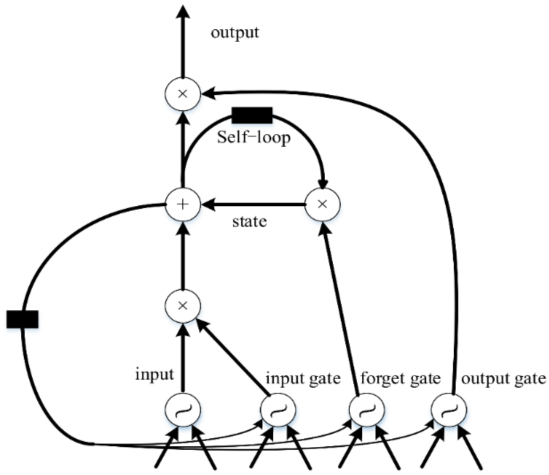

Figure 2. Building blocks of LSTM recurrent network of a "cell"

All gates can also receive additional information from the cell's state. A pivotal element is the state unit, featuring a linear self-loop. A forget gate unit $f_i^{(t)}$ for a given time, step *t* and cell *i* govern the self-loop weight, assigning a value between 0 and 1 through a sigmoid unit.

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) \quad (3)$$

In an LSTM network, the hidden layer vector, $h^{(t)}$ comprises all the LSTM cell outputs and denotes current input data captured as a numerical vector. The forget gate variables and Wf represent the equivalent biases, input weights, and recurrent weights connections. LSTM cell memory is dynamically adjusted by a self-loop gate controlled by external input $f_i^{(t)}$.

The LSTM network is a powerful tool for processing sequential data and is commonly used for various applications such as language translation, natural language processing, speech recognition, and image captioning. The LSTM network is a type of RNN designed to address the vanishing gradient problem and make it more effective for handling long-term dependencies. The structure of the weights and biases of an LSTM can be examined by providing the declaration of the hidden state and the total number of neurons in the LSTM.

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right) \quad (4)$$

Where b, U, and W stand for the biases, input weights, and recurrent weights of the LSTM cell, respectively. The external input gate unit generates a gating value between 0 and 1, whereas the forget gate employs a sigmoid unit $g_i^{(t)}$ and uses its criteria to accomplish the same objective.

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right) \quad (5)$$

The output gate $q_i^{(t)}$, which also employs a sigmoid unit for gating, can disable the output $h_i^{(t)}$ of the LSTM cell.

$$h_i^{(t)} = \tanh\left(s_i^{(t)}\right) q_i^{(t)} \quad (6)$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right) \tag{7}$$

Its input weights, recurrent weights, and biases are composed of the three parameters: $bo, Uo,$ and W^o. One of the options, denoted by the three gates of the i^{th} unit, may accept the cell state $s_i^{(t)}$ and its weight as an extra input. RNNs have specialized variants like LSTM networks and gated recurrent unit (GRU) [29] networks that effectively manage long-term dependencies in sequential data by selectively retaining relevant information through "gates". This allows the LSTM networks to overcome the vanishing gradient problem, making them powerful tools for handling long-term dependencies and sequential data. There are other variants of RNNs, such as fully RNN (FRNN), Independent RNN (IndRNN), and Bidirectional RNNs, each with unique characteristics and applications. Long-term dependencies may be learned by LSTM networks more quickly than by simple recurrent designs, as has been shown. The capacity to acquire long-term dependencies was tested on simulated data sets and complex sequence processing problems that achieved state-of-the-art performance.

4. RESULT AND DISCUSSION

To study weather classification, we employ a Kaggle dataset comprising 1,125 unobstructed images from various online sources, such as Flickr, Getty Images, Yahoo, Google Images, and more. These images have been categorized into four groups: sunrise, cloudy, rainy, and shining. Figure 3 illustrates the distribution of images within the dataset, with sunrise images accounting for 357, cloudy images for 300, shining images for 253, and rainy images for 215. The images are in PNG format and have been resized to a tensor of dimensions 50×50×3, a standard format for machine learning models to process visual data. This dataset offers a valuable resource for researchers and developers to explore and enhance weather recognition algorithms.

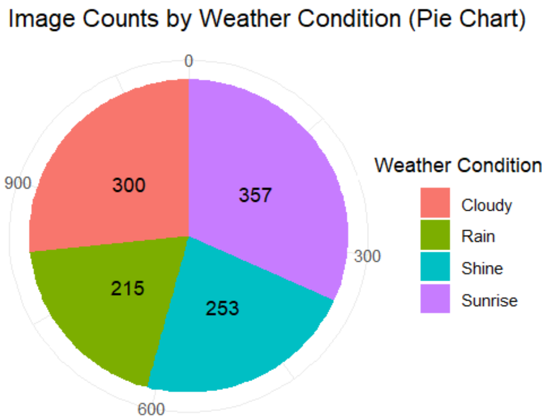

Figure 3. Statistics of the dataset used for the research purpose

The dataset was split into training (80%) and validation (20%) sets for model development. The implementation utilizes the Python programming language and uses the computational power of Kaggle's GPU kernels to ensure efficient training and analysis. This choice provided access to high-performance hardware resources, enabling faster processing and model optimization. Images of weather conditions of different zones make up the picture data used as the model's training and testing input. Figure 4 shows a graph that compares the total loss of the LSTM model to its total validation loss. The graph shows how the total loss of the FCN-LSTM model gradually increases over time. Specifically, the loss starts to rise significantly after the 30th epoch, indicating that the model's performance is not as accurate as it was earlier.

The temporal and geographical range covered by the FCN-LSTM model type is broad. Deep spatial information is encoded, and natural language strings are produced using a convent (encoder) and an LSTM (decoder). It is possible to represent sequential data at different times using LSTM. On the other hand, the total validation loss's value is continuously rising and falling. The value increases relative to the other points as the tidal validation graph approaches epoch 30. In this instance, the total loss is smaller than the total loss due to validation. The proposed model's overall accuracy against the total validation accuracy graph is shown in Figure 5. This graph demonstrates that the inclusive accuracy of the proposed model varies somewhat, with a noticeable rise in volatility occurring before the number approaches thirty. The total validation accuracy of our model graph exhibits a notable early decline, followed by a rebound up to a specific moment. The validation accuracy of the proposed model gradually increases until it reaches the 40-epoch mark. After that, the validation accuracy remains steady but does not match the overall accuracy.

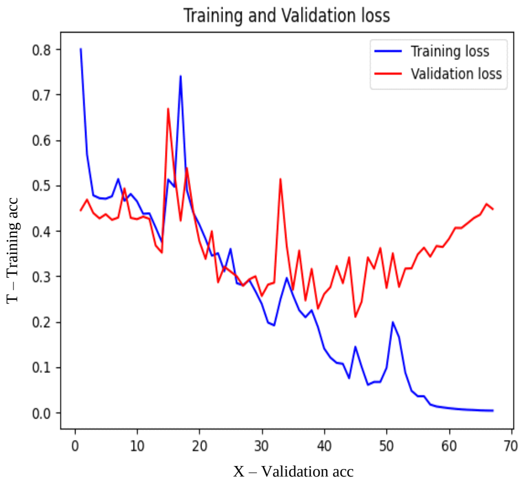

Figure 4. FCN-LSTM training and validation loss curve

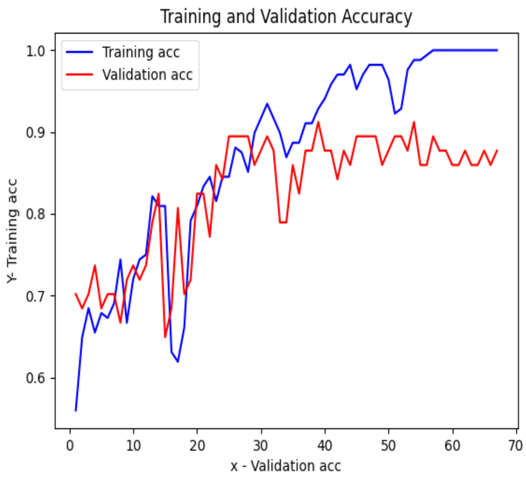

Figure 5. FCN-LSTM training and validation accuracy curve

Figure 6 compares the accuracy of weather classification using different methods: CNN, LSTM, a hybrid CNN-LSTM, and a proposed model. Specifically, the CNN yields 84.56%, the LSTM yields 82.1%, the hybrid CNN-LSTM yields 91.55%, and the proposed model yields 96.88%. The results show that the proposed model achieves the highest accuracy, more than a 5% improvement over the existing methods. Also, the validation accuracy has reached the value of 91.22%. This figure surpasses the current models and comes with complete automation support. The accuracy of weather forecasts using the FCN-LSTM image-based model is also superior. This comparison demonstrates the exceptional performance of the proposed model in weather classification tasks.

Figure 7 highlights a confusion matrix for weather classification, revealing the performance of the proposed FCN-LSTM model. The matrix displays the predicted and actual classes, highlighting the model's average accuracy of 96.88%, precision, recall, and F1-score for each weather category. The matrix demonstrates the model's ability to correctly identify sunny and cloudy conditions, which is crucial for defining ideal climatic zones.

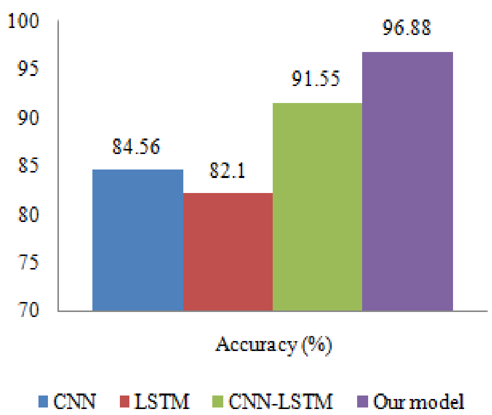

Figure 6. Accuracy comparison with existing methods for weather classification

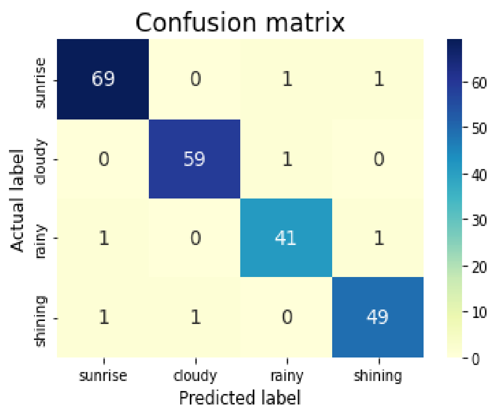

Figure 7. Confusion matrix for weather classification

Figure 8 compares the MSE of different methods for weather classification, including CNN, LSTM, a hybrid CNN-LSTM model, and a proposed model. The CNN is used for weather classification, while the LSTM network learns sequences within weather data, uncovering hidden patterns and their relationships. The hybrid CNN-LSTM model combines the strengths of both architectures, and the proposed model aims to improve existing weather classification and forecasting methods. For the proposed model, the output classification generated is very close to the actual weather condition values; it is also observed through the MSE value, which is very low at 7.11.

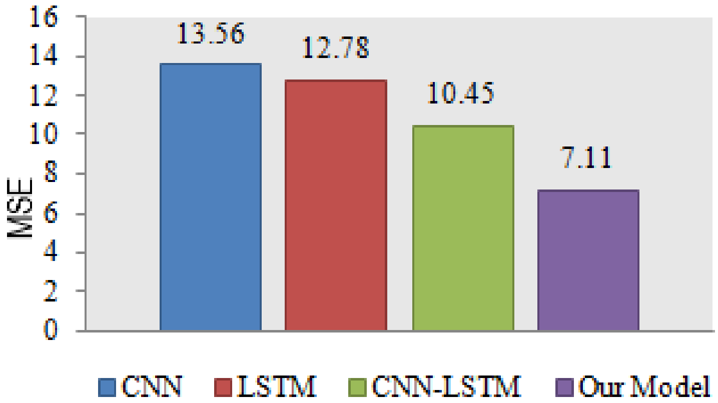

Figure 8. MSE comparison with existing methods for weather classification

5. CONCLUSION AND FUTURE WORKS

This research aims to create a fully automated image-based weather forecasting and classification system. It uses deep learning, FCN, and LSTM to identify the type of weather and predict future weather conditions. It offers a method that uses the VGG16 and FCN-LSTM to reliably extract the correlated features and uses these features for weather forecasting based on input images. This technology is very helpful to the government, various organizations, and farmers in planning their activities according to current weather classification and possibilities. Once the model is instantiated in the real world, its automation initiates warnings automatically when it identifies and predicts the weather conditions in a specific area. The accuracy data indicates that the model's 96.88% accuracy is higher than the accuracy of current systems and that the number of false alarms has significantly decreased as MSE is 7.11 and is found to be better than that of existing systems. The system currently uses real-time image input and employs deep-learning methods to enhance precision and reduce development times. Future iterations of the system could integrate diverse data sources like real-time sensor data like temperature, humidity, satellite imagery, and weather reports to offer a more holistic understanding. Refined forecasts could localize weather patterns for specific areas and provide updates at finer time scales hourly, even minute-by-minute. Interpretable models explaining predictions and quantifying their uncertainty could be developed, boosting user trust and aiding decision-making. Data augmentation techniques and transfer learning from larger datasets could be explored to improve performance in regions with limited data.

REFERENCES

[1] X. Chen *et al.*, "Short-term and local rainfall probability prediction based on a dislocation support vector machine model using satellite and in-situ observational data," *IEEE Access*, pp. 1–1, 2024, doi: 10.1109/ACCESS.2019.2913366.
[2] A. Barszcz, J. A. Milbrandt, and J. M. Thériault, "Improving the explicit prediction of freezing rain in a kilometer-scale numerical weather prediction model," *Weather and Forecasting*, vol. 33, no. 3, pp. 767–782, 2018, doi: 10.1175/WAF-D-17-0136.1.
[3] A. Subashini, "Advanced weather forecasting prediction using deep learning," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, no. 8, pp. 939–945, 2019, doi: 10.22214/ijraset.2019.8139.
[4] P. Mohan and K. K. Patil, "Survey on crop and weather forecasting based on agriculture related statistical data," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 2320–9801, 2017, doi: 10.15680/IJIRCCE.2017.
[5] F. Sheikh, S. Karthick, D. Malathi, J. S. Sudarsan, and C. Arun, "Analysis of data mining techniques for weather prediction," *Indian Journal of Science and Technology*, vol. 9, no. 38, 2016, doi: 10.17485/ijst/2016/v9i38/101962.
[6] W. C. Woo and W. K. Wong, "Operational application of optical flow techniques to radar-based rainfall nowcasting," *Atmosphere*, vol. 8, no. 3, 2017, doi: 10.3390/atmos8030048.
[7] J. Li and X. Luo, "A study of weather-image classification combining VIT and a dual enhanced-attention module," *Electronics (Switzerland)*, vol. 12, no. 5, 2023, doi: 10.3390/electronics12051213.

[8] R. U. Rani, J. Kakarla, and B. Sundar, "Weather image classification using EfficientNet and dual attention block," in *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing, ICSTSN 2023*, 2023, pp. 1–4, doi: 10.1109/ICSTSN57873.2023.10151564.

[9] S. Mittal and O. P. Sangwan, "Classifying weather images using deep neural networks for large scale datasets," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, pp. 337–343, 2023, doi: 10.14569/IJACSA.2023.0140136.

[10] M. F. Naufal and S. F. Kusuma, "Weather image classification using convolutional neural network with transfer learning," *AIP Conference Proceedings*, vol. 2470, 2022, doi: 10.1063/5.0080195.

[11] H. Xiao, F. Zhang, Z. Shen, K. Wu, and J. Zhang, "Classification of weather phenomenon from images by using deep convolutional neural network," *Earth and Space Science*, vol. 8, no. 5, 2021, doi: 10.1029/2020EA001604.

[12] Y. Bai, J. Wang, X. Wang, and H. Li, "The summary of deep learning in the field of weather forecast research," *Journal of Physics: Conference Series*, vol. 1646, no. 1, 2020, doi: 10.1088/1742-6596/1646/1/012035.

[13] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Analysis and Applications*, vol. 24, no. 1, pp. 343–366, 2021, doi: 10.1007/s10044-020-00898-1.

[14] X. Ma, C. Fang, and J. Ji, "Prediction of outdoor air temperature and humidity using Xgboost," *IOP Conference Series: Earth and Environmental Science*, vol. 427, no. 1, 2020, doi: 10.1088/1755-1315/427/1/012013.

[15] J. Frnda, M. Durica, J. Rozhon, M. Vojtekova, J. Nedoma, and R. Martinek, "ECMWF short-term prediction accuracy improvement by deep learning," *Scientific Reports*, vol. 12, no. 1, 2022, doi: 10.1038/s41598-022-11936-9.

[16] X. Dong, S. Deng, and D. Wang, "A short-term power load forecasting method based on k-means and SVM," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 11, pp. 5253–5267, 2022, doi: 10.1007/s12652-021-03444-x.

[17] B. Shao, D. Song, G. Bian, and Y. Zhao, "Wind speed forecast based on the LTSM neural network optimized by the firework algorithm," *Advances in Materials Science and Engineering*, vol. 2021, 2021, doi: 10.1155/2021/4874757.

[18] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognition Letters*, vol. 141, pp. 61–67, 2021, doi: 10.1016/j.patrec.2020.07.042.

[19] J. A. Weyn, D. R. Durran, and R. Caruana, "Can machines learn to predict weather? Using deep learning to predict gridded 500-hpa geopotential height from historical weather data," *Journal of Advances in Modeling Earth Systems*, vol. 11, no. 8, pp. 2680–2693, 2019, doi: 10.1029/2019MS001705.

[20] D. S. Roy, "Forecasting the air temperature at a weather station using deep neural networks," *Procedia Computer Science*, vol. 178, pp. 38–46, 2020.

[21] S. Ravuri *et al.*, "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, no. 7878, pp. 672–677, Sep. 2021, doi: 10.1038/s41586-021-03854-z.

[22] L. Espeholt *et al.*, "Deep learning for twelve hour precipitation forecasts," *Nature Communications*, vol. 13, no. 1, 2022, doi: 10.1038/s41467-022-32483-x.

[23] Q. Fu, D. Niu, Z. Zang, J. Huang, and L. Diao, "Multi-stations' weather prediction based on hybrid model using 1D CNN and Bi-LSTM," *Chinese Control Conference, CCC*, vol. 2019-July, pp. 3771–3775, 2019, doi: 10.23919/ChiCC.2019.8866496.

[24] V. Chole, M. Thawakar, M. Choudhari, S. Chahande, S. Verma, and A. Pimpalkar, "Enhancing heart disease risk prediction with GdHO fused layered BiLSTM and HRV features: a dynamic approach," *Biomedical Signal Processing and Control*, vol. 95, 2024, doi: 10.1016/j.bspc.2024.106470.

[25] A. Pimpalkar and J. R. Raj R, "MBiLSTMGloVe: embedding GloVe knowledge into the corpus using multi-layer BiLSTM deep learning model for social media sentiment analysis," in *Expert Systems with Applications*, vol. 203, 2022.

[26] Y. Son, X. Zhang, Y. Yoon, J. Cho, and S. Choi, "LSTM–GAN based cloud movement prediction in satellite images for PV forecast," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 9, pp. 12373–12386, 2023, doi: 10.1007/s12652-022-04333-7.

[27] S. Sengar and X. Liu, "Ensemble approach for short term load forecasting in wind energy system using hybrid algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 5297–5314, Nov. 2020, doi: 10.1007/s12652-020-01866-7.

[28] I. Mansoury, D. El Bourakadi, A. Yahyaouy, and J. Boumhidi, "A novel wind power prediction model using graph attention networks and bi-directional deep learning long and short term memory," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 6, pp. 6847–6854, 2023, doi: 10.11591/ijece.v13i6.pp6847-6854.

[29] A. Pimpalkar and J. R. Raj, "A bi-directional GRU architecture for the self-attention mechanism: an adaptable, multi-layered approach with blend of word embedding," *International Journal of Engineering and Technology Innovation*, vol. 13, no. 3, pp. 251–264, 2023, doi: 10.46604/ijeti.2023.11510.

BIOGRAPHIES OF AUTHORS

Nilesh Shelke is an assistant professor in computer science and engineering Department of Symbiosis Institute of Technology, Nagpur. He is M.Tech. in computer science and engineering from RTM Nagpur University. He is M.Phil. in computer science and also acquire Ph.D. in computer science and engineering from S.G.B. Amravati University, Amravati. He is Microsoft Certified Solution Developer and has more than 25 years of experience of imparting IT training which includes learners from different streams, faculties and Microsoft Certifications to corporate employees. His area of research includes natural language processing, image processing, machine learning, deep learning. He has published patents, copyrights and technical articles in the renowned journals. He has also authored the books' introduction to machine learning, an introduction to artificial intelligence and machine learning, knowledge representation and artificial intelligence: ML and DL. He can be contacted at email: nilesh.shelke@sitnagpur.siu.edu.in.

Sudhanshu Maurya is a visiting post-doctoral researcher at the Department of Engineering, University of Luxembourg, Luxembourg, Europe. He has completed Post-Doctoral Research at School of Computer and Communication Engineering, Universiti Malaysia Perlis (UniMAP), Kangar, Perlis, Malaysia. He has completed Doctor of Philosophy (Ph.D.), thesis entitled: "A study on security and privacy aspects of mobile cloud computing." He has completed M.Tech. computer science and engineering (Secured 1st position) from the School of Engineering and Technology, Hemvati Nandan Bahuguna Garhwal University (A Central University), India. He is currently working as associate professor – CSE, Symbiosis Institute of Technology, Nagpur Campus, a constituent of Symbiosis International (Deemed University), Maharashtra, India. He has 11 years of experience and authored/coauthored 125+ research papers indexed in SCOPUS/Web of Scopus, 16 patents, 5 course book, and 2 edited books. He can be contacted at email: dr.sm0302@gmail.com.

Rupali Ithape is currently working as an assistant professor in the Electronics and Communication Engineering (ECE) Department at the School of Engineering and Sciences at MIT-ADT University, Pune. She earned her Master of Technology from Bharati Vidyapeeth Deemed University, Pune (India), in 2014. She received her bachelor of engineering from Pune University, Nagpur, in 2006. She has 17 years of academic experience. She has published one research paper in the International Journal of Engineering Research and Technology. She had one National Patents on her name. She is a Life Member of IETE. Her research interest includes image processing and deep learning. She can be contacted at email: rupali.ithape@mituniversity.edu.in.

Zarina Shaikh is currently working as an assistant professor in the Computer Science and Engineering (CSE) Department at Dr. D. Y Patil Institute of Technology, Pune. She is pursuing her PhD in computer science and engineering from Shri Jagdishprasad Jhabarmal Tibrewala University (JJTU) University. She received her master of engineering from Dr. D. Y. Patil College of Engineering, Akurdi, Pune, in 2013, her bachelor of engineering from Shivaji University in 2003, and her diploma in computer engineering from Maharashtra State Board of Technical Education, Mumbai, in 2000. She has 17 years of academic experience. She has published more than 20+ research articles in international journal and conferences proceedings. She had national patents on her name. She is a life member of ISTE. Her research interest includes machine learning, cloud computing. She can be contacted at email: shaikhzarina14@gmail.com.

Rachna Somkunwar has completed her PhD in computer science and engineering from Shri Jagdishprasad Jhabarmal Tibrewala University (JJTU) in 2019 and M.Tech. and B.E. in computer engineering from Nagpur University. She has 18 years of teaching experience. Her research interest includes pattern matching, AI, and data mining. She has authored 40+ research articles and conference papers in reputed international journals and conferences. Her 2 patents are granted, 13 patents are published, and 24 copyright registered. Her 2 book chapters have been published. She has authored and co-authored 4 books. She is currently associated with the Department of Computer Engineering of Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, as an associate professor. She can be contacted at email: rachnasomkunwar12@gmail.com.

Amit Pimpalkar is currently working as an assistant professor in the Computer Science and Engineering (AIML) Department at Shri Ramdeobaba College of Engineering and Management, Nagpur. He earned his PhD in computer science and engineering from the Sathyabama Institute of Science and Technology, Chennai, in 2023. He received his Master of Technology from Shri Ram Institute of Technology, Jabalpur, in 2013, his Bachelor of Engineering from Nagpur University, Nagpur, in 2005, and his diploma in computer technology from Maharashtra State Board of Technical Education, Mumbai, in 2002. He has 19 years of academic and industrial experience. He has published more than 80 research articles in International Journal and Conference proceedings. He had Ten National and an International patent on his name. He is a life member of ISTE, IAENG, ICSES, CSTA, and IEEE. He is an active reviewer for many national and international conferences and journals, including IEEE Access, Wiley, Hindawi, Elsevier, Heliyon, and the Journal of Sensors. His research interest includes data mining, machine learning, and NLP. He can be contacted at email: amit.pimpalkar@gmail.com.