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

# Deep Learning for Anomaly Detection in Spatio- Temporal Maharashtra Weather Data: A Novel Approach with Integrated Data Cleaning Techniques

**Kunal Kulkarni[1], Yashashree Mahale[1], Nida Khan[1], Nandhini K.[1], Shilpa Gite[2]**

**Submitted:** 17/11/2023 **Revised:** 29/12/2023 **Accepted:** 09/01/2024

**Abstract**: Maharashtra, located in the western part of India, experiences diverse climatic conditions owing to its vast geographical expanse. Seasonal patterns, such as the monsoon rains and dry summers, significantly impact the weather dynamics. This research includes primary data of Maharashtra State Monthly Dataset spanning from 2001 to 2022. Central to our approach is the integration of the expectation maximization optimization technique for data cleaning, addressing the challenges of noise and inconsistencies within the dataset. The primary objective is to enhance the robustness and accuracy of the weather data, laying a foundation for more reliable anomaly detection. Leveraging state-of-the-art algorithms such as One-Class SVM, Isolation Forest, LSTM Autoencoders, and Autoencoders, the research scrutinizes their efficacy in identifying anomalies within the complex temporal and spatial patterns inherent to Maharashtra's climate. The integrated data cleaning approach emerges as a novel aspect of this research, revealing its positive impact on refining the deep learning models' performance. Visualizations aid in intuitively understanding the detected anomalies and their implications for weather analysis. The results and discussion sections meticulously compare the outcomes of each algorithm, offering insights into their strengths and limitations. This approach provides a robust framework for anomaly detection in Maharashtra's weather data, enabling enhanced climate trend analysis, early detection of irregularities, and improved decision-making for disaster preparedness and resource allocation in the face of changing weather patterns.

***Keywords****— climate, deep learning, LSTM Autoencoders, spatio-temporal*

## 1. Introduction

Climate plays a major role in shaping a region's socio-economic and ecological landscape, making accurate and timely weather forecasts [1] crucial for industries such as agriculture, disaster management, energy production, and transportation. Maharashtra, a diverse and populous state in India, is no exception to the profound impacts of weather variability and extreme events on its society and economy. Maharashtra experiences a wide range of weather conditions due to its geographical expanse, which spans the western coastline of the Arabian Sea to the central Deccan Plateau and the eastern Ghats.

Maharashtra's economy heavily relies on agriculture, making it imperative to provide accurate predictions of rainfall and temperature for informed decision-making by farmers. Unpredictable weather patterns brought on by climate change have an effect on water resources and agriculture. Weather forecasts are important to the energy sector because they affect wind energy generation and hydropower [2]. Furthermore, weather data are needed for infrastructure development and urban planning to be resilient to extreme events. Given these justifications, it is clear that spatiotemporal analysis is necessary.

Spatio-temporal data refers to information that varies both in space (location) and time (period) [3]. It depicts the dynamic character of phenomena at various times and locations across space. Maps, satellite photos, sensor readings, weather reports, and other data can all be used to depict spatiotemporal data. A comprehensive understanding of weather phenomena allows for a more accurate representation of both its temporal dynamics and geographical distribution. Deep learning presents a viable substitute for comprehending and forecasting these changes with greater accuracy since it can comprehend intricate and dynamic relationships in data. Using deep learning techniques to analyze spatiotemporal weather data may be a solution to these needs. [4]. With its ability to capture intricate patterns and dependencies in data, deep learning can improve weather predictions significantly. Deep learning models, like convolutional neural network (CNN), recurrent neural network (RNN) and Long Short-Term Memory (LSTM) are capable of capturing the intricated

[1]*Artificial Intelligence and Machine Learning Department, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India.*
*2 Symbiosis Centre for Applied AI, Symbiosis International (Deemed University), Pune, India.*
*kulkarnikunal63gmail.com, yashashree125@gmail.com, nidak6478@gmail.com, nandhinik2@gmail.com*
*shilpagite15@gmail.com*
*Corresponding Author-Dr. Shilpa Gite*

patterns and dependencies in data, resulting in improved accuracy and reliability of weather predictions

## 2. Related Work

An overall review of the work in spatiotemporal analysis provides us information about various machine learning and deep learning approaches used in research which includes Long Short-Term Memory model (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network-LSTM (CNN-LSTM), Spatio-Temporal Graph Convolutional Network (STGCN) and Geographic Semantic Temporal Hypergraph Convolutional Network etc.

In [5], the authors proposed LSTM with a new architecture for spatiotemporal analysis for real life prediction using ocean dataset which consists of three different dataset which was collected through sensors and attributes such as current velocity, temperature, and dissolved oxygen were considered.

The authors in [6] implemented deep learning models such as CNN, Graph CNN, RNN, LSTM, AE/SAE, RBM, and Seq2Seq models for Spatio Temporal data mining (STDM) on spatiotemporal including predictive learning, classification, estimation, representation learning and inference, anomaly detection, and others. A variety of domains, such as transportation, on-demand services, human mobility, climate and weather, location-based social networks (LBSN), crime analysis, and neuroscience, were represented by the applications of deep learning techniques for STDM.

Chunrui Wu et al. [7] summarized outlier detection for spatiotemporal data as data collection has become easier detecting the outlier becomes essential for real world application such as geophysical exploration and geological disaster monitoring. The proposed model used various clustering, classification and deep learning methods and also reviews spatial and temporal outlier mining methods for different types of datasets.

Authors in [8] proposed a system for spatio-temporal prediction of climate and other environmental features using deep learning. They divided the spatiotemporal signal in stochastic spatial coefficients and fixed temporal bases which rebuilds irregular distributed measurements and uses any regression algorithm for spatial prediction of stochastic coefficients. Framework establishes a positive approach in identifying temporal, spatial, and spatio-temporal dependencies in data that is simulated and real-world data.

Convolutional Long Short-Term Memory (ConvLSTM) and the Graph Convolutional Network (GCN) was used in [9] to predict hourly meteorological, wildfire, remote sensing satellite, and ground-based sensor data combined to provide spatiotemporal PM2.5 in Los Angeles County. It created images of the dense meteorological graphs using unsupervised graph representation learning algorithms for ConvLSTM's input.

A novel deep graph-based structure is presented in [10] for solving the STLF problem. CNN and GRU are used for feature extraction and feature understanding implemented on an actual dataset of Shiraz, Iran. The new technique is compared with CNN, MLP, and KNN.The authors concluded that new graph-based structure was more efficient as compared to regular techniques used as it provides more accuracy.

Climatic factors for cereal crop yields are studied in [11] with the data of the eastern plateaus zone for 6 regions. The Mann Kendall test and pooled OLS regression method is used for determining the suitable climate for crops. There were no apparent trends in this region's cereal yields whereas temperature and precipitation had different effects on cereal yields. A weather– yield relationship is observed which can be studied and evaluated further.

In [12] the author proposed HetSPGraph with LSTM model for drought forecasting. It identified spatial correlations in drought data that changes with time and spatio-temporal correlation is obtained which is used as an input for temporal drought forecasting. HetSPGraph model was observed to be a flexible approach for analyzing multivariate TS.

The first paper that used Autoformer and LogSparse Transformer is [13,21] for wind forecasting as an updated function for GNN, these techniques were compared with Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) models and it was concluded that the novel approach with Transformer and autoformer outperformed LSTM and MLP models.

The authors of [14] used multiple iterative imputation using autoencoder-based LSTM for forecasting concentration on air pollutants, which also included an LSTM autoencoder for identifying and eliminating outliers from the dataset followed by forecasting PM2.5 concentration using a multivariate LSTM. A comparison was carried out between various models like 1D convolutional neural network (CNN), gated recurrent unit (GRU) and long short-term memory (LSTM) and it was concluded that imputation for anomaly removal helped to increase the accuracy for forecasting air pollution.

With the help of real data from Cyber-physical system (CPS) considering controlled and physical attacks as anomalies the authors of [15] proposed an Approximate Projection Autoencoder (APAE) which uses autoencoders to create two defenses for attacks that include one novel approach for improvising robustness by adversarial impact by using optimizing latent representations by better reconstruction output. It was concluded that combining defenses improves attack identification.

In [16] author proposed two different autoencoders for combination learning of node and attributes embedding, to detect anomalies on attributed networks on several real-world datasets, it was found that AnomalyDAE performs better than the state- of-the-art techniques at the moment.

The authors of [17] concentrated on the problem of detecting the Anomaly for Indoor Air Quality (IAQ) which was previously tried to be tackled with the help of machine learning algorithms for anomaly detection which had several drawbacks this was rectified by using deep learning approach for anomaly detection which consists of a hybrid model with a combination of Long short-term memory (LSTM) and autoencoder This approach outperformed the previous models and obtained an outstanding accuracy.

In paper [18] author used Recurrent neural network (RNN) which is called Bidirectional Long Short-Term Memory (BLSTM) in order to find the quality intervals based on forecasts to obtain high coverage probability and narrow interval widths for wind speed interval prediction in order to predict the characteristics of wind energy.

The literature review covers various machine and deep learning algorithms that can be used for anomaly detection for spatio- temporal as well as other data. Researchers have examined the efficacy of these algorithms across diverse fields, including industrial systems, finance, cybersecurity, and healthcare. In order to improve anomaly detection accuracy, the review also covered hybrid approaches, parameter tuning, and optimization techniques. Furthermore, it's possible that current research has concentrated on solving issues with these algorithms' interpretability, scalability, and imbalanced datasets. For the most recent developments and discoveries in the field of anomaly detection using these algorithms, it is therefore imperative to refer to the most recent literature.

## 3. Proposed Methodology

*A. Outline of the methodology*

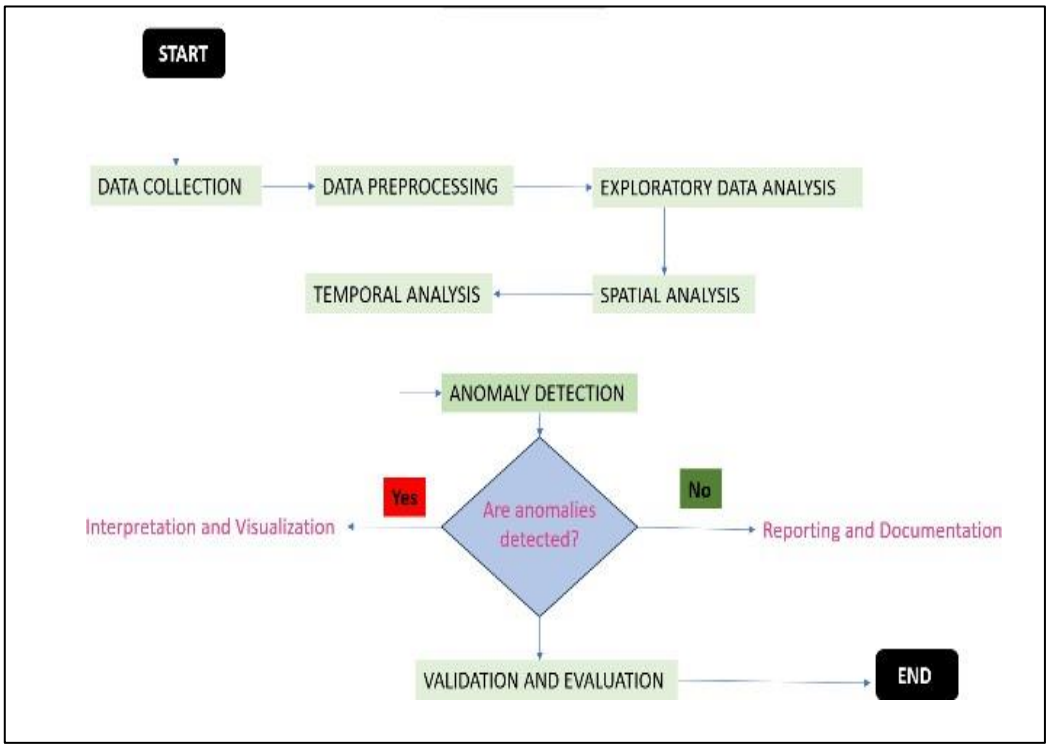


**Fig 1:** Proposed Flow

As described in the Figure 1, spatiotemporal analysis of Maharashtra weather dataset is conducted as part of this research project with the goal of identifying subtle patterns and trends in the meteorological data in both the space and time dimensions. To identify irregularities and outliers within this primary dataset hence leveraging the power of deep learning. The utilization of deep learning methodologies holds promise for capturing intricate relationships and subtle variations in weather patterns, enabling us to discern anomalies that may signify significant deviations from the norm. This study contributes to the evolving field of Spatio-temporal analysis by integrating advanced machine learning techniques to enhance our understanding of Maharashtra's weather dynamics and to fortify our ability to identify anomalous events that may have far-reaching implications for climate monitoring and prediction. Furthermore, upon the successful identification and removal of outliers through our deep learning-based anomaly detection approach, we lay the foundation for more robust and accurate weather forecasting. By eliminating spurious data points that could otherwise distort predictive models, we enhance the reliability of the dataset, thus facilitating more precise and efficient forecasting of weather conditions in Maharashtra.

### *B. Dataset*

The dataset used in this research was obtained from the India Meteorological Department (IMD), Pune, a reputable government organization in India. IMD operates a network of weather stations, observatories, and instruments throughout India, including the state of Maharashtra. The dataset encompasses a comprehensive collection of weather-related parameters, recorded on a monthly basis, and spans multiple stations across the state of Maharashtra. It spans a duration from the year 2001 to 2021 and then some data for 2022 and 2023 and comprises a comprehensive collection of meteorological parameters recorded across all weather stations throughout the state of Maharashtra. These parameters include rainfall, temperature, wind speed, evaporation, sunshine hours, gull intensity, dust levels, and storm occurrences etc. along with latitudes and longitudes of the respective stations. Table I shows the list of parameters included in the dataset. Figure 2 demonstrates the analysis for weather distribution by district.

**Table 1:** Features of Dataset

| | |
|---|---|
| INDEX | Index Number of station |
| MN | Month |
| MMAX | Mean Maximum Temp (deg C) |
| MMIN | Mean Minimum Temp (deg C) |
| NO | No of Observations |
| TMRF | Total Rainfall in the Month (mm) |
| MWS | Mean Wind Speed (kmph) |
| MEVP | Mean Evaporation (mm) |
| MSSH | Duration of Sunshine (hrs.) |
| LATITUDE | Latitude of the Station |
| LONGITUDE | Longitude of the Station |

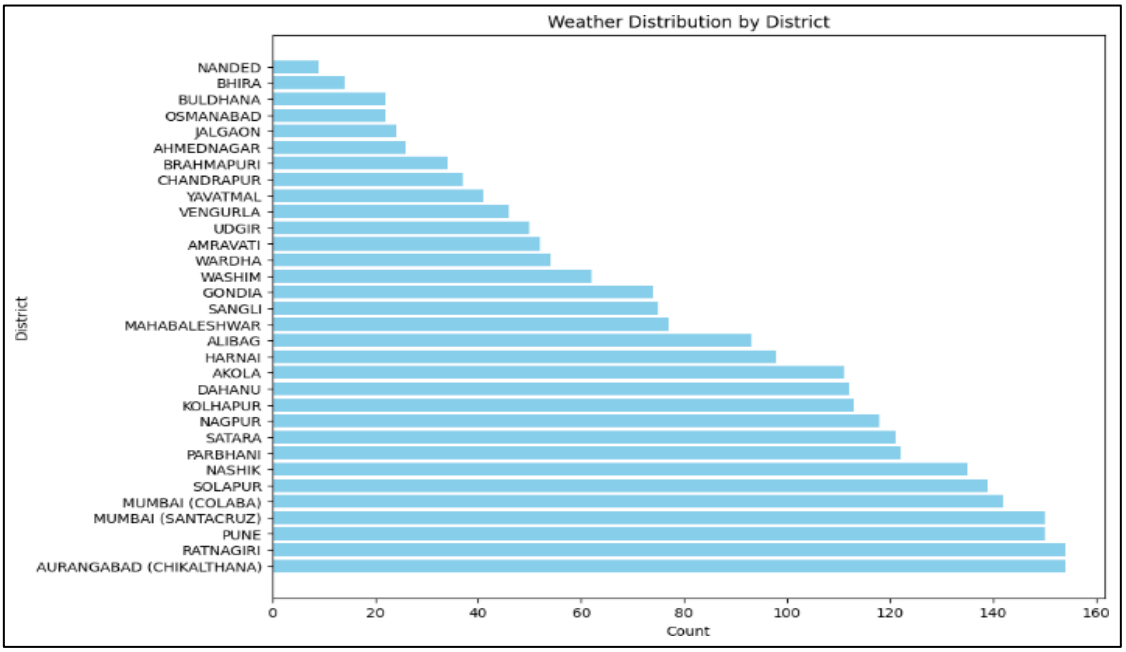


**Fig 2:** Weather Distribution by District

In accordance with ethical and legal standards, appropriate permissions and access rights were obtained from IMD, Pune through their data supply portal https://dsp.imdpune.gov.in/ to utilize their meteorological dataset for this research.

Additionally, any necessary data preprocessing and quality control procedures were implemented to ensure the dataset's integrity and suitability for analysis.

*C. Preprocessing*

Considering Temporal weather data of Maharashtra state, there were 49 features which had null values which needed to be tackled. As this is a time-series data the following optimization imputations are implemented.

*Optimization Imputation Technique: Expectation Maximization*

Expectation-Maximization (EM) is a statistical algorithm commonly used for handling missing or null values in time-series data. The algorithm alternates between two main steps: Expectation and Maximization. When missing data are present in a statistical model, the EM algorithm attempts to estimate the parameters with the highest likelihood. Below are the steps employed in the algorithm:

1. Initialization: The process begins with an initial set of parameter estimates for the time-series model. Additionally, missing values are initialized with suitable starting values or estimates.

$\theta_0$ represents the time-series model's initial parameter estimates.

2. Expectation Step (E-step): Based on the observed data and the current parameter estimates, the algorithm determines the expected values of the missing data at the E-step. Specifically, the observed data is used as a conditional probability distribution to compute missing values. The missing values are updated subsequent to the conditional probabilities being calculated.

$Q(\theta|\theta_t) = E_{missing}$ [log $L(\theta;observed,imputed)|observed, \theta_t]$

Here, Q - expected log-likelihood function, θ - parameter vector, and

$\theta_t$ - current parameter estimates.

Given the current parameter estimates and the observed data, the expectation is taken for the missing data.

$E(missing|observed,\theta_t)$

3. Maximization Step (M-step): In the M-step, the likelihood function is maximized while accounting for both the imputed and observed data. The time-series model's parameters are updated by combining imputed and observed data. In order to find parameter values that maximize the likelihood of both observed data and imputed data, the optimization problem will be solved as part of this step.

$\theta t+1=\arg\max\theta\ Q(\theta|\theta t)$

4. Convergence Check: Convergence is checked by assessing the change in parameter estimates between consecutive iterations. If the change falls below a

predefined threshold or a maximum number of iterations is reached, the algorithm is considered to have converged. Otherwise, the E-step and M-step are repeated. Check if or if a predefined maximum number of iterations is reached, where $\epsilon$ is a small threshold.

$$\|\theta t+1-\theta t\|<\epsilon \quad (4)$$

5. Iterative Refinement: Steps 2 to 4 are iteratively performed until convergence is achieved. The iterative nature of the algorithm allows for the refinement of parameter estimates and imputed values over successive iterations. Additionally, imputation techniques of mean, mode, median were also utilized and a clean dataset is observed.

*D. Feature Selection*

The feature space's volume increases exponentially with the number of features which can lead to the curse of dimensionality, making it challenging to build accurate and efficient models, especially when the number of samples is limited. The dataset contained 49 features for all the districts in the Maharashtra state. Enabling feature selection significantly improve computational efficiency by reducing the number of features without sacrificing predictive performance. To assess the interdependence among features, a correlation matrix was computed for the entire dataset. The correlation matrix provided a pairwise correlation coefficient for each pair of features, ranging from -1 to 1. Values closer to 1 indicate a strong positive correlation, values -1 indicate a strong negative correlation, and values 0 indicate no linear correlation. This coefficient quantifies the linear relationship between variables. To focus on highly correlated features, a threshold of 0.5 was set. Features with absolute correlation coefficients greater than this threshold were considered highly correlated as shown in the Figure 3. The final set of features was determined by computing the correlation matrix for the selected features and retaining only those features with significant correlations. Table 2 gives the list of features obtained after feature selection.

**Table 2:** Selected Features

| | |
|---|---|
| MMAX | Mean Maximum Temp (deg C) |
| MMIN | Mean Minimum Temp (deg C) |
| HMAX | Highest Maximum Temp (deg C) |
| LMIN | Lowest Minimum Temp (deg C) |
| NO | No of Observations |
| TMRF | Total Rainfall in the Month (mm) |
| HVYRF | Heaviest 24 hrs. Rainfall (mm) |
| Humidity | Humidity |
| MWS | Mean Wind Speed (kmph) |
| MEVP | Mean Evaporation (mm) |
| MSSH | Duration of Sunshine (hrs.) |
| LATITUDE | Latitude of the District |
| LONGITUDE | Longitude of the District |
| INDEX | Index number of the station |
| DISTRICT | District Name |

Along with the selected columns, we also include district names, latitude and longitude of the regions.

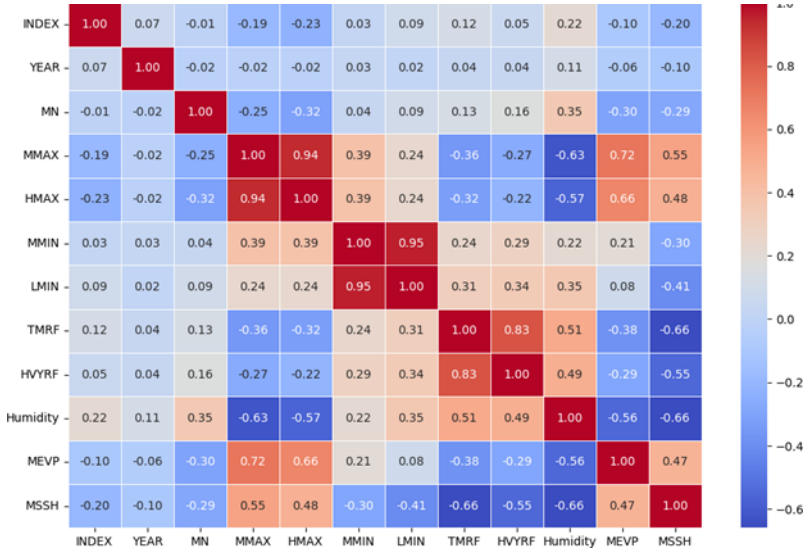


**Fig 3:** Heatmap of selected Features

*E. Proposed System*

*1. Models Used:*

Leveraging the capabilities of machine learning and deep learning, a diverse set of algorithms are applied to the dataset, including One Class SVM, Isolation Forest, Autoencoders and Long Short-Term Memory based Autoencoders (LSTM) to model complex spatial and temporal relationships in the weather data. One-Class SVM is specifically used to identify anomalies by learning a hyperplane that divides normal instances from the rest. Isolation Forest is used because it partitions the dataset randomly, which allows it to isolate anomalies. By encoding and reconstructing the data, autoencoders a neural network-based technique captures intricate patterns and relationships in the data, with anomalies being detected based on high reconstruction errors. Furthermore, complex spatial and temporal relationships in the weather data are modeled using Long Short-Term Memory (LSTM) based Autoencoders, which makes them especially useful for capturing anomalies in spatiotemporal patterns. Combining these algorithms allows for a thorough approach to anomaly detection, taking into account the dataset's temporal and spatial complexities. The proposed system for the anomaly detection in the spatio-temporal data as explained in the Figure

4. The working of these models is explained below:

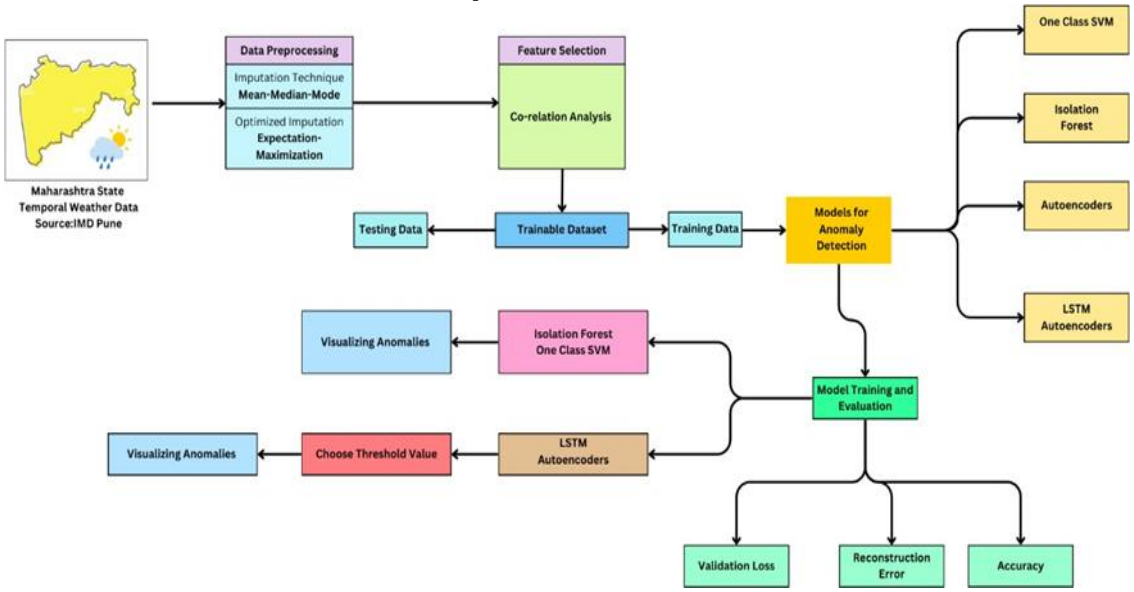


**Fig 4:** Proposed Methodology

1.1. One-Class SVM:

A particular kind of support vector machine intended for anomaly detection is the One-Class Support Vector Machine (OCSVM) [19]. OCSVM is trained solely on normal data, in contrast to traditional SVMs that are trained on both normal and abnormal data. Its objective is to identify a model of typical behavior and identify any departure from this model as an anomaly. One-

Class SVM (OCSVM) algorithm works well since it can represent the typical patterns in the data and identify deviations as anomalies. Learning a decision boundary that captures the data's typical behavior is necessary for One-Class. SVM (OCSVM) to detect anomalies in spatiotemporal meteorological data. The following are the crucial steps:

i. Training: OCSVM is trained solely with typical spatiotemporal meteorological data examples. It gains the ability to define a hyperplane, or a collection of hyperplanes, that spans most of the typical patterns found in the data. The form and flexibility of the decision boundary are determined by the selection of the kernel function, also known as the radial basis function or RBF kernel, and its parameters, such as gamma. One important hyperparameter in OCSVM is the nu parameter. It manages the trade-off between identifying data points as outliers and having a smooth decision boundary. It displays an upper bound for the fraction of margin errors and a lower bound for the fraction of support vectors.

ii. Testing and Anomaly Detection: The model can give new instances of anomaly scores once it has been trained. The degree of deviation between the instance and the learned normal behavior is indicated by the anomaly score's magnitude. For every data point, the decision function yields a decision value. Generally speaking, an outlier (anomaly) is indicated by a negative value, while an inlier (normal) is indicated by a positive value.

iii. Threshold Setting: A threshold is set to categorize instances as normal or anomalous based on the anomaly scores on a validation set. Anomalies are those instances where the scores are higher than the cutoff.

1.2. Isolation forest:

Another popular algorithm for anomaly detection is isolation forest [20], which is an ensemble-based anomaly detection algorithm that is particularly effective for high-dimensional datasets, such as spatiotemporal data. The foundation of the algorithm is the notion that anomalies are simpler to identify in the feature space than typical occurrences. The following describes how to use Isolation Forest for spatiotemporal weather data anomaly detection:

i. Isolation Tree Construction: A random subsample of the data (with replacement) is used for every tree in the ensemble. To divide the data, a random feature is chosen at each node of the tree. Until each instance is isolated in its own leaf node, the data is recursively divided into two subsets. It is anticipated that anomalies will be isolated faster than typical cases.

ii. Model Training: Separate constructions are made of several isolation trees. The accuracy of the anomaly scores tends to increase with the number of trees in the ensemble.

iii. Anomaly Score Calculation: The average path length in the tree ensemble is used to compute an anomaly score for every instance. Shorter average path lengths suggest that the instance is more likely to be an anomaly because they make it simpler to isolate.

iv. Anomaly Detection: A threshold is utilized to identify anomalies. Anomalies are defined as instances with anomaly scores greater than the cutoff. Every instance is predicted by the model to be either an outlier (an anomaly) or an inlier (normal).

1.3. Autoencoders

Neural network architectures known as autoencoders are useful for anomaly detection and unsupervised learning. Autoencoders are a useful tool for learning a compact representation of normal patterns in spatiotemporal data. They can be used to identify anomalies by evaluating deviations from this learned representation. The following describes the use of autoencoders for spatiotemporal weather data anomaly detection:

i. Model Architecture: The input data is compressed into a lower-dimensional representation (latent space) by the autoencoder's encoder component. It has one or more dense layers with ReLU-like activation functions. The compressed representation is used by the decoder to reconstruct the input data. It is an inverted mirror of the encoder's structure.

ii. Training: Only typical feature selected spatiotemporal weather data instances are used to train the autoencoder. The normal patterns are first encoded and then decoded by the autoencoder. Utilizing the mean squared error loss function and a suitable optimizer (such as Adam), compile the autoencoder model.

iii. Reconstruction error: After training, use the autoencoder to reconstruct the input data. For every instance, calculate the mean squared error between the input data and the reconstructed output.

iv. Anomaly Detection: Establish a threshold for the error in reconstruction. Reconstruction errors greater than this cutoff are regarded as anomalies.

Depending on whether an instance's reconstruction error is above or below the threshold, categorize it as abnormal or normal.

1.4. LSTM-Autoencoders

A network architecture mainly used for sequence-to-sequence learning is (LSTM autoencoder, which can be used for anomaly detection in spatiotemporal data.

Below is an overview of the proposed model how anomaly detection using LSTM autoencoders operates:

i. Encoder: The temporal data input sequence is processed by the encoder portion of the LSTM autoencoder, which then compresses it into a fixed-size latent representation. The encoder uses LSTM layers to identify patterns and temporal dependencies in the input sequence.

ii. Latent Representation: A compressed, lower-dimensional representation of the input sequence is called the latent representation. It records the data's key characteristics and temporal patterns.

iii. Decoder: The LSTM autoencoder's decoder component uses the latent representation to try and piece together the original input sequence. The decoder also employs LSTM layers to produce a sequence that, in theory, ought to resemble the input sequence.

iv. Training: The weather data is used to train the autoencoder. The mean squared error (MSE) between the input and the reconstructed output must be kept to a minimum. In order to reduce the reconstruction

v. error, the input sequences are fed into the model during the training phase, and the weights are updated.

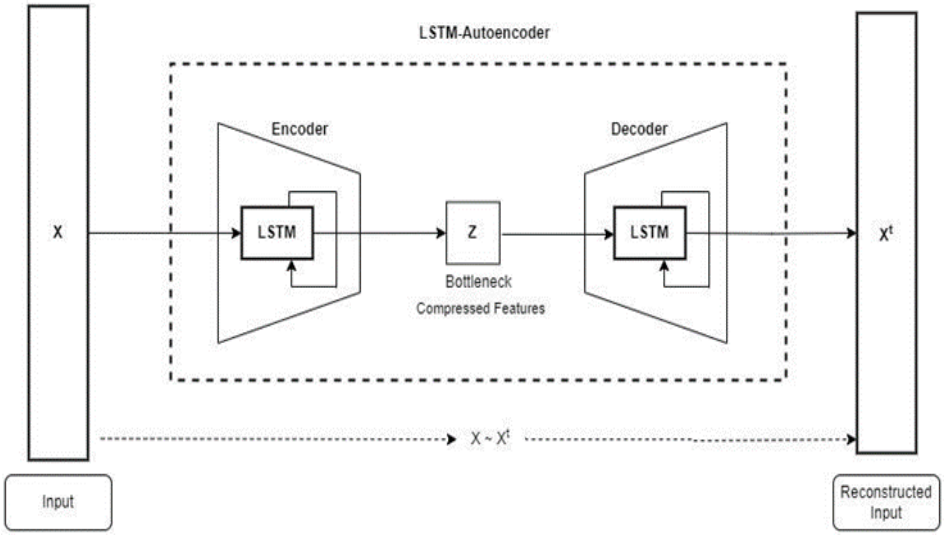


**Fig 5**: LSTM-Autoencoders

Anomaly Detection: After training the LSTM autoencoder on the weather data, it can be used for anomaly detection in the following way:

i. Reconstruction Error: Sequence reconstruction can be done with the model once it has been trained. The mean squared error (MSE) between the input sequences and them reconstructed counterparts is computed in order to identify anomalies. Higher reconstruction error sequences are regarded as anomalies.

ii. Thresholding: A threshold is set on the reconstruction error. Anomalies are identified by comparing the MSE values with a predefined threshold. Data points with MSE above the threshold are considered anomalies.

iii. Anomaly Detection: The anomalies are visualized with respect to various features, such as 'Index', 'YEAR,', 'MN', 'District,' 'Latitude,' and 'Longitude.'

In summary, the above section concluded different machine learning and deep learning models that we implemented, Autoencoders and LSTM autoencoders are based on neural networks, whereas One-Class SVM and Isolation Forest are classic machine learning algorithms. One-Class SVM, an approach that uses a learned hyperplane to identify anomalies; Isolation Forest, an unsupervised method that isolates anomalies by random partitioning; Autoencoders, neural network-based unsupervised models that encode and reconstruct data, with anomalies detected through high reconstruction errors; and LSTM Autoencoders, which use recurrent neural networks for sequence data and are especially useful in time series anomaly detection, are just a few of the algorithms used in anomaly detection. Since each algorithm has

unique properties, the appropriateness of each depends on the properties of the data and the particular requirements for detection.

## 4. Experimental Results

The study explored various machine learning and deep learning including One-Class SVM, Isolation Forest, Autoencoders and LSTM-Autoencoders. The Figure 6 shows the visualization of loss and validation loss obtained by performing various epochs which was carried out to obtain minimum loss. Following this, the performance of each model and the number of anomalies detected are respectively are shown in the below Table 3.

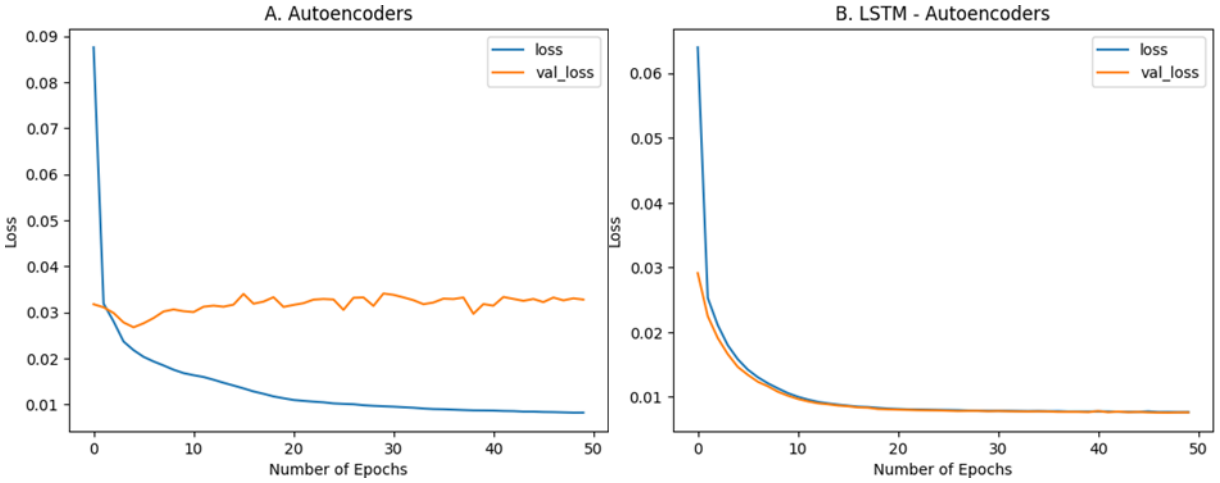


**Fig 6: (A).** Epochs Vs Loss for Autoencoders, **(B).** Epochs Vs Loss for LSTM Autoencoders

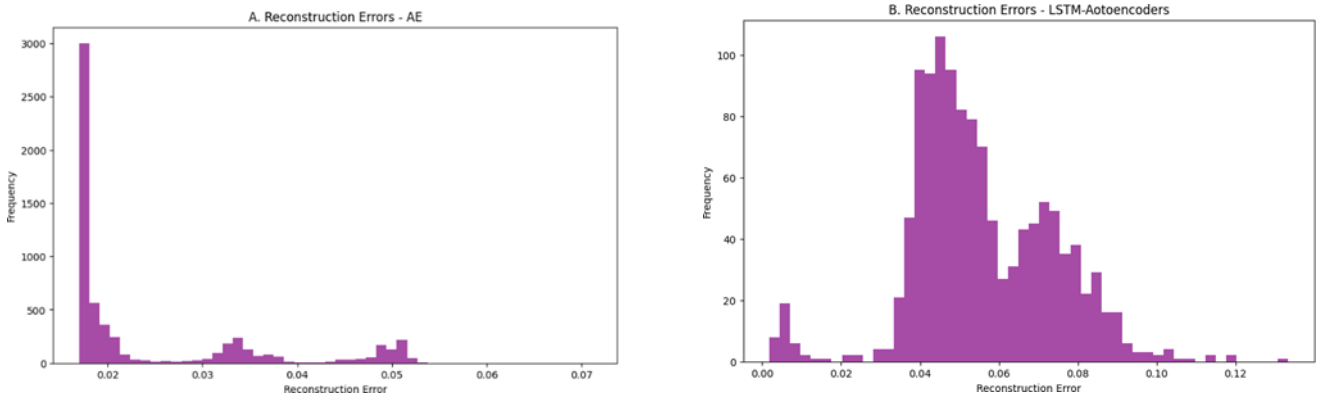


**Fig 7:** (A.) Reconstruction Errors for Autoencoders, (B). Reconstruction Errors for LSTM Autoencoders

In Figure 7 (A.) The reconstruction errors are visualized using the histogram which explains that the ranges of reconstruction errors for Autoencoders is between 0.02 to 0.05 whereas in (B) the reconstruction errors of LSTM-AE ranges from 0.0015 to

0.12 which concludes that the visualization in fig(A) is right skewed while the visualization in fig(B) is equally distributed resulting to obtain appropriate reconstruction error which can further be used to decide the value of threshold.

The Table 3 shows the comparative analysis of various models like One class SVM, Isolation Forest, Autoencoder and LSTM autoencoders where parameters like validation loss mean construction error, threshold and number of anomalies detected were noted. One class SVM and Isolation Forest did not detect the appropriate number of anomalies. On comparison it was observed that Autoencoders detected anomalies but was sensitive to noisy data therefore anomalies detected were not accurate whereas LSTM autoencoder had detected anomalies accurately where threshold was considered as 0.5638 and 0.5733 respectively

**Table 3:** Comparative Analysis of applied Models

| Model Used | Description | Validation Loss | Mean Reconstruction Error | Chosen Threshold | No. of Anomalies Detected | Challenges Encountered |
|---|---|---|---|---|---|---|
| One-class SVM | Learns a decision boundary around a single class of data, identifying anomalies as points that fall outside the boundary. | N/A (No labelled data for metrics) | N/A | Not Required | 303 | Choice of kernel function is complex, very less anomalies detected |
| Isolation Forest | Isolates anomalies by randomly partitioning the data into smaller groups and iteratively removing the easiest instances to isolate. | N/A (No labelled data for metrics) | N/A | Not Required | 304 | Unsupervised technique hence may not be effective |
| Autoencoders | Learn to encode and decode data, identifying anomalies as data that cannot be accurately reconstructed using the learned encoding and decoding processes | 0.0342 | 0.02401 | 0.05638 | 1731 | sensitive to noisy data and is not effective for detecting subtle anomalies |
| LSTM Autoencoders | Learns a compressed representation of normal data, flagging anomalies as data that cannot be accurately reconstructed using the learned representation. | 0.0080 | 0.0075 | 0.05733 | 528 | Requires careful hyperparameter tuning and not suitable for real-time anomaly detection. |

.

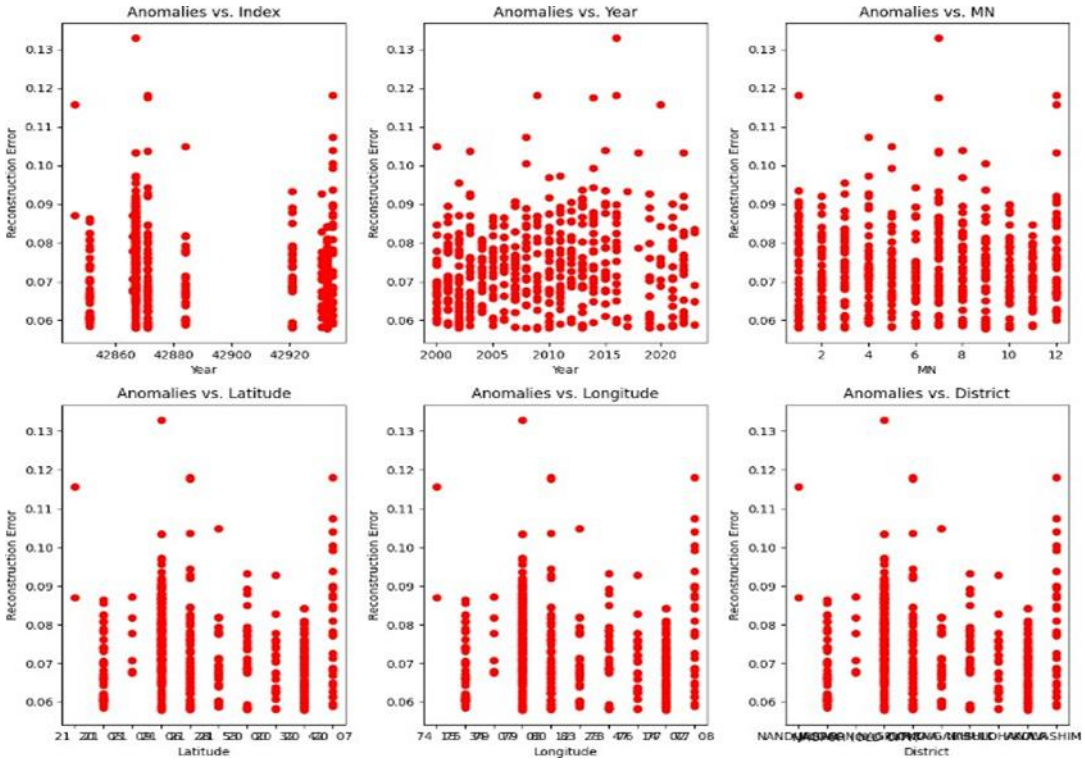


**Fig 8:** Anomalies detected with respect to Index, Year, Month, Latitude, Longitude, Districts

Figure 8 visualizes the representation of anomalies with scatter plots with respect to Index, Year, Month Latitude, Longitude and Districts. These scatter plots contribute to a holistic understanding of anomaly distribution across spatial locations, district and time. Whereas Figure 9 is a 3D representation of the

respective features which helps understanding the spread of anomaly. These 3D plots offer advance visualizations of anomalies providing multidimensional perspective among district, spatial coordinates and temporal dynamics.

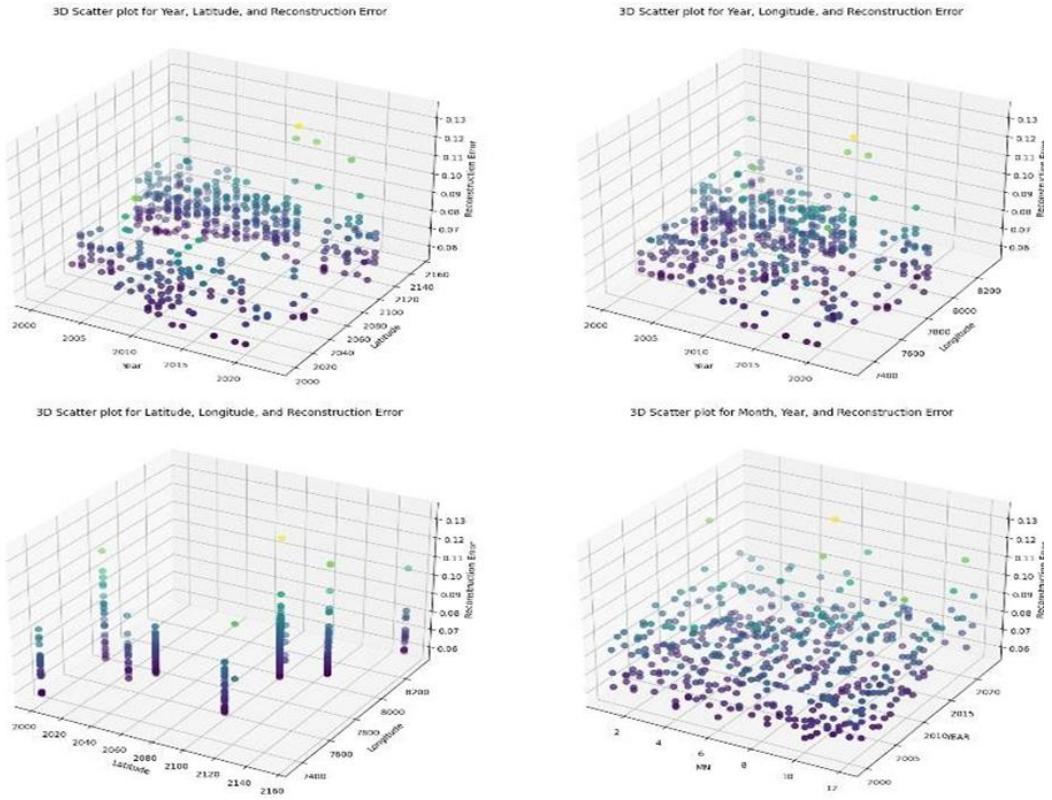


**Fig 9:** 3D representation of the respective reconstruction errors

Anomalies with Spatio Temporal Features using Autoencoders:

| | YEAR | MN | District | Latitude | Longitude | Anomaly Count |
|---|---|---|---|---|---|---|
| 0 | 2000.0 | 1.0 | MUMBAI (COLABA) | 18 54 | 72 49 | 10 |
| 1 | 2014.0 | 6.0 | GONDIA | 21 28 | 80 12 | 11 |
| 2 | 2014.0 | 3.0 | UDGIR | 18 04 | 77 07 | 12 |
| 3 | 2014.0 | 4.0 | MUMBAI (COLABA) | 18 54 | 72 49 | 13 |
| 4 | 2014.0 | 4.0 | NAGPUR | 21 06 | 79 03 | 14 |
| .. | ... | ... | ... | ... | ... | ..... |
| 746 | 2007.0 | 9.0 | UDGIR | 18 04 | 77 07 | 1450 |
| 747 | 2007.0 | 10.0 | NAGPUR | 21 06 | 79 03 | 1451 |
| 748 | 2007.0 | 10.0 | PUNE | 18 32 | 73 51 | 1452 |
| 749 | 2007.0 | 10.0 | UDGIR | 18 04 | 77 07 | 1453 |
| 750 | 2023.0 | 6.0 | NAGPUR | 21 06 | 79 03 | 1454 |

[751 rows x 6 columns]

**Fig 10 (A):** Anomalies detected by autoencoders

Anomalies with Spatio Temporal features using LSTM-Autoencoders:

| | YEAR | MN | District | Latitude | Longitude | Anomaly Count |
|---|---|---|---|---|---|---|
| 0 | 2000.0 | 1.0 | GONDIA | 21 28 | 80 12 | 1 |
| 1 | 2011.0 | 12.0 | NAGPUR | 21 06 | 79 03 | 1 |
| 2 | 2013.0 | 3.0 | NASHIK | 20 00 | 73 47 | 1 |
| 3 | 2013.0 | 3.0 | NAGPUR | 21 06 | 79 03 | 1 |
| 4 | 2013.0 | 2.0 | NAGPUR | 21 06 | 79 03 | 1 |
| .. | ... | ... | ... | ... | ... | ... |
| 1450 | 2005.0 | 9.0 | NAGPUR | 21 06 | 79 03 | 1 |
| 1451 | 2005.0 | 9.0 | JALGAON | 21 03 | 75 34 | 1 |
| 1452 | 2005.0 | 7.0 | RAIGARH | 21 53 | 83 23 | 1 |
| 1453 | 2005.0 | 7.0 | NAGPUR | 21 06 | 79 03 | 1 |
| 1454 | 2023.0 | 6.0 | NAGPUR | 21 06 | 79 03 | 1 |

[455 rows x 6 columns]

**Fig 10 (B):** Anomalies detected by LSTM-Autoencoders

There were around 751 anomalies detected by autoencoders as shown in the Figure 10 (A) whereas the number of anomalies detected by LSTM-Autoencoders were 455. The difference in the anomalies found by the two models indicates that autoencoders are sensitive to finding noisy or odd patterns in the dataset. The observed variations in anomaly detection results are probably due to the unique traits and advantages of LSTM-Autoencoder model, particularly in capturing spatial and temporal dependencies. Additional examination of the characteristics of the anomalies and the performance indicators of each model may shed light on how well each one captures and identifies anomalous occurrences in the data.

## 5. Conclusion

In conclusion, this research presents a holistic approach to Spatio-temporal analysis and anomaly detection in the Maharashtra weather dataset, with a specific focus on enhancing forecasting capabilities. Through meticulous feature selection and data pre- processing, we curated a refined dataset that captures the intricacies of meteorological dynamics in the region. Leveraging the power of Long Short-Term Memory (LSTM) based autoencoders, our anomaly detection methodology demonstrated effectiveness in identifying and eliminating outliers. Improved accuracy and resilience show that the successful removal of anomalies cleared the path for the creation of more dependable forecasting models. In order to show the spatial distribution of anomalies and give a more nuanced understanding of how weather irregularities manifest in different parts of

Maharashtra, we conducted visualizations at the district level. The inclusion of station-level temporal visualizations in this spatial analysis allows for a comprehensive examination of the temporal patterns associated with detected anomalies. Incorporating visualizations not only facilitates the interpretation of detected anomalies but also enhances the intuitive understanding of the spatiotemporal dynamics of the dataset. The clear and understandable depiction of anomalous weather events provided by these graphics can be beneficial to meteorologists, policymakers, and stakeholders alike. The data acquired from this research may improve weather forecasting, guarantee more precise forecasts, and support the building of climate resilience in the face of changing environmental challenges.

## 6. Future Work

This study provides opportunities for more research in a number of areas. First, examining the integration of additional data sources such as satellite imagery or remote sensing could enhance the precision and detail of the spatiotemporal analysis. Techniques for anomaly detection can be made better. Adjusting hyperparameters, exploring ensemble methods, and experimenting with different deep learning architectures can all help improve the anomaly detection model's sensitivity and specificity. Analyzing comparisons with other states or regions could yield important information about how generalizable the anomaly detection and forecasting techniques created for Maharashtra are.

## References


[1] Jaseena, K. U., and Binsu C. Kovoor. "Deterministic weather forecasting models based on intelligent predictors: A survey." Journal of King Saud University-Computer and Information Sciences 34.6 (2022): 3393-3412.

[2] Li, Zhenhui, and Shuchen Xiang. "A design of new wind power forecasting approach based on IVMD-WSA-IC-LSTM model." Journal of Engineering and Applied Science 70.1 (2023): 91.

[3] Kumari, Sushma, et al. "Spatio-temporal analysis of air quality and its relationship with COVID-19 lockdown over Dublin." Remote Sensing Applications: Society and Environment 28 (2022): 100835.

[4] Ma, Minbo, et al. "HiSTGNN: Hierarchical spatio-temporal graph neural network for weather forecasting." *Information Sciences* 648 (2023): 119580.

[5] O'Donncha, Fearghal, et al. "A spatio-temporal LSTM model to forecast across multiple temporal and spatial scales." Ecological Informatics 69 (2022): 101687.

[6] Sharma, Arun, Zhe Jiang, and Shashi Shekhar. "Spatiotemporal data mining: A Survey." arXiv preprint arXiv:2206.12753 (2022).

[7] Wu, JT Chunrui, and Junfeng Tian. "Spatio-temporal outlier detection: A survey of methods." International Journal of Frontiers in Engineering Technology 2.1 (2020).

[8] Amato, Federico, et al. "A novel framework for spatio-temporal prediction of environmental data using deep learning." Scientific reports 10.1 (2020): 22243.

[9] Muthukumar, Pratyush, et al. "PM2. 5 Air Pollution Prediction through Deep Learning Using Multisource Meteorological, Wildfire, and Heat Data." Atmosphere 13.5 (2022): 822.

[10] Ganjouri, Mahtab, et al. "Spatial-temporal learning structure for short-term load forecasting." IET Generation, Transmission & Distribution 17.2 (2023): 427-437.

[11] Benmehaia, Amine M., Noureddine Merniz, and Amine Oulmane. "Spatiotemporal analysis of rainfed cereal yields across the eastern high plateaus of Algeria: an exploratory investigation of the effects of weather factors." Euro-Mediterranean Journal for Environmental Integration 5 (2020): 1-12.

[12] Balti, Hanen, et al. "Spatio-temporal heterogeneous graph using multivariate earth observation time series: Application for drought forecasting." Computers & Geosciences 179 (2023): 105435.

[13] Bentsen, Lars Ødegaard, et al. "Spatio-temporal wind speed forecasting using graph networks and novel Transformer architectures." Applied Energy 333 (2023): 120565.

[14] Narkhede, Gaurav, et al. "Novel MIA-LSTM Deep Learning Hybrid Model with Data Preprocessing for Forecasting of PM2. 5." Algorithms 16.1 (2023): 52.

[15] Goodge, Adam, et al. "Robustness of autoencoders for anomaly detection under adversarial impact." Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 2021.

[16] Fan, Haoyi, Fengbin Zhang, and Zuoyong Li. "Anomalydae: Dual autoencoder for anomaly detection on attributed networks." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

[17] Wei, Yuanyuan, et al. "LSTM-autoencoder-based anomaly detection for indoor air quality time-series data." IEEE Sensors Journal 23.4 (2023): 3787-3800.

[18] Saeed, Adnan, et al. "Hybrid bidirectional LSTM model for short-term wind speed interval prediction." IEEE Access 8 (2020): 182283-182294.

[19] Erfani, Sarah M., et al. "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning." Pattern Recognition 58 (2016): 121-134.

[20] Xu, Hongzuo, et al. "Deep isolation forest for anomaly detection." IEEE Transactions on Knowledge and Data Engineering (2023).

[21] Hudnurkar, Shilpa, et al. "Multivariate Time Series Forecasting of Rainfall Using Machine Learning." Artificial Intelligence of Things for Weather Forecasting and Climatic Behavioral Analysis, edited by Rajeev Kumar Gupta, et al., IGI Global, 2022, pp. 87-106. https://doi.org/10.4018/978-1-6684-3981-4.ch007