Univariate Time Series Anomaly Detection on AirQuality Data

Deepthy Prasad
Stockholm University
Sweden
depr5821@student.su.se

Swathi Rao Hampapura Sripada
Stockholm University
Sweden
swha6754@dsv.su.se

***Abstract*— We are witnessing air pollution as one of the major environmental health problems that need to be attended and monitored. With the advancements in technology, the use of sensors has helped in keeping track of the quality of the air. So, this tends to generate a huge amount of data that becomes almost impossible to intervene manually and understand. It becomes important to keep track of the scenarios where there are unusual readings to take necessary actions against them. Such erroneous data, which is marked as an anomaly or an outlier, need to be detected in an automated way [1]. This paper implements three different methods - prediction, statistics and clustering, using various Machine learning models - Autoregressive Integral Moving Average (ARIMA), Long Short Term Memory (LSTM) and Density-based spatial clustering of applications with noise (DBSCAN), to detect these anomalies and visualise them as graphs.**

Keywords—air quality, machine learning, LSTM, ARIMA, DBSCAN, anomaly detection

I. Introduction

With the latest technological advances, the amount of data collected over time in many research areas is significant. When such data have been collected over a period that are correlated in time, it becomes time series data. Such data can be analyzed and studied to conclude using various data mining techniques. And one among them is the detection of outliers/anomalies present in the data. Many researchers and practitioners are interested in outlier detection, which concentrates on time series data mining [2]. Most of the studies conducted can be observed that it relies on simulation techniques and mathematical equations for understanding how the time series data has evolved. And they have been represented by statistical methods and algorithms of machine learning [3].

An anomaly/outlier is a data point which can be differentiated from the rest of the data as it seems to have an uncommon pattern or trend, or seasonal aspect from most data points. It can also be considered in a way that its statistical properties do not match the rest of the series [4].

Many systems have been built with a huge amount of data collected over time, and any changes in the behaviours of the systems can lead to serious consequences or failure. And detecting anomalies and keeping track of them becomes a critical task. So, in any area with time series data, it is important to separate the outliers and pay attention to them before it leads to any underlying issue. Time series data can be either multivariate or univariate. When it comes to univariate time series data, Autoregressive Integral Moving Average (ARIMA) model is often used for forecasting [3]. Freeman et al. proposed a forecasting method for time series data related to air quality with the help of a deep learning model consisting of recurrent neural network (RNN) and Long short-term memory (LSTM) [5]. Density-based spatial clustering of applications with noise (DBSCAN) is a widely used clustering algorithm in anomaly detection. The paper by [6] shows that the DBSCAN model has high performance compared to the K-means model. Also, another study [7] proves that DBSCAN can spot unusual data points that deviate from the actual data distribution.

II. Purpose

The aim was to study how data collected over a period contributing to the time series data could be analyzed and made used to detect the unusual behaviour of the data. Such time series data should be pre-processed and analysed differently before using it in the models and architectures. The purpose is to use anomaly detection models to process the time series data and analyse how the outliers could be detected. The models selected here are LSTM, ARIMA and DBSCAN. It is demonstrated how each model was uniquely designed to suit the requirements, and uncommon behaviours in the collective dataset are detected.

III. Method

A. Data

The data for detecting anomalies was an air quality dataset taken from Kaggle [8]. It had 15 features consisting of five metal oxide chemical sensor data installed in a highly polluted area. The level of CO, Benzene, Total Nitrogen Oxide (NOx), Non-Metanic Hydrocarbons and Nitrogen Dioxide(NO2) could be seen. Its readings were taken as an hourly averaged response consisting of about 9357 observations. Readings were recorded from March 2004 to February 2005 with a frequency of every 1 hour a day. As it is a study where univariate time series anomaly detection is implemented, only one column is selected out of 15 features. To get an overlook of how the data trend is, observations from one month span of time were taken for a few of the sensor data and plotted, which can be seen in Figure 1. It can be seen how the outliers exist in the collected dataset.

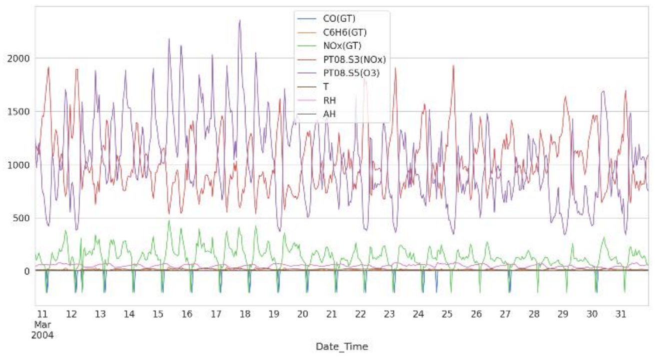

Figure 1 Air quality data for one month (10/03/04 - 31/03/04) showing readings for CO, NOx, Temperature, Humidity etc

B. *Anomaly detection methods*

Various Machine Learning(ML) techniques can be used for anomaly detection in different ways. In this project, we have implemented three ML models with three different approaches for univariate time series anomaly detection [4].

The first approach is statistical, where the mean and standard deviation measures of the input values are used to set a threshold and all the data points outside the threshold are marked as anomalies. The ARIMA model is used in this project to implement this. The second approach is a predictive modelling approach using a supervised learning method, and we chose LSTM to perform this. In this, the input data is split into train and test and create a model to predict anomalies using train data and prediction will be applied to test data. This approach allows us to measure the error rate in the forecast and thereby understand how well the time series anomaly detection model performs. The last approach is clustering using any unsupervised learning techniques where the data points can be clustered and categorize the non-clustered data points as anomalies. This approach can be implemented with ML unsupervised techniques, which are robust to outliers. DBSCAN was found to be a good clustering model which is robust to outliers [7] and the third approach is carried out with it.

C. *PreProcessing*

Once the dataset was chosen, we decided to take one feature -*Relative Humidity* out of all the 15 features forming the dataset for analysis and detection of the anomalies. It could be seen that the co-relation of this feature with other features (sensor data of the harmful components) was more than 50%. So, this was chosen to detect the anomalies. We can see from Figure 2 that there are observations which fall in the negative scale of -200, and the dataset was designed to replace the missing values with this value. Among many existing ways to handle such missing values, we chose to replace it with the previous value/row entry to not disturb the sensor information collected and not reduce the data instances.

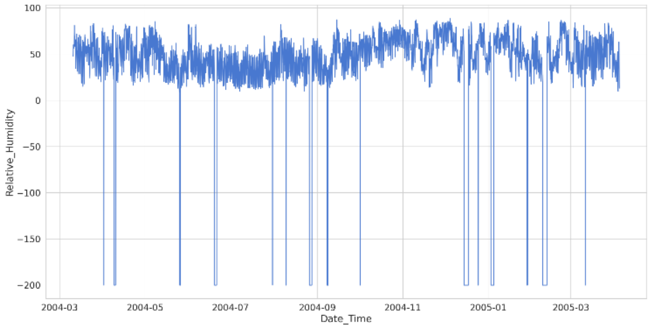

Figure 2 Missing values in Relative Humidity feature

As we are implementing 3 different anomaly detection models, the preprocessing also has been done depending on the model design explained below:

In the case of the LSTM model, the data has been normalized using a standard scalar normalization technique and transformed to form the dataset, as shown in Figure 3. To prepare the data for the LSTM model and its layers within, it was reshaped accordingly.

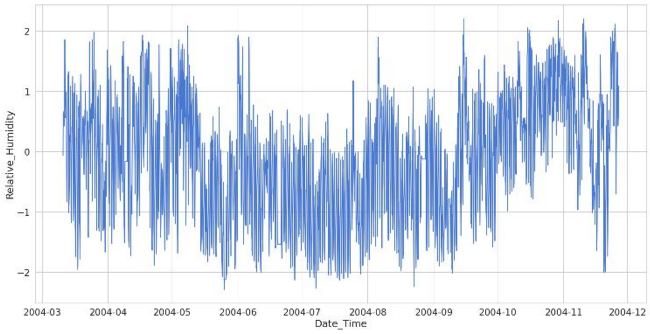

Figure 3 Standardized input data for the LSTM model

ARIMA model is a widely used time series anomaly detection method, and it requires stationary input data. The stationarity check Augmented Dickey-Fuller (ADF) is utilized in the pre-processing, and it was discovered that the input was not stationary. To make the input data steady, a simple temporal shift technique was employed. The created input is shown in Figure 4:

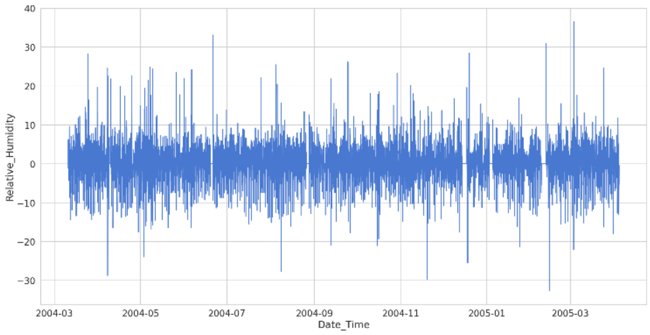

Figure 4 Stationary input data for the ARIMA model

Since DBSCAN is a clustering algorithm, no special preprocessing was necessary. So the input data after imputation have been used in further steps. The input data in graphical form is shown in Figure 5:

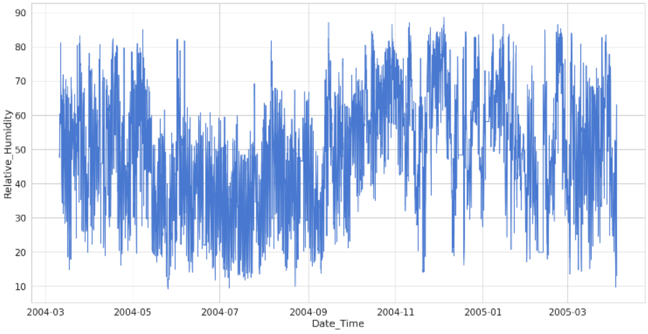

Figure 5 Input data for the DBSCAN model

D. Models Designed

a) LSTM Model

The model was designed with LSTM and Autoencoder. An autoencoder is good at learning the coding of unlabeled data and ignores insignificant data [9]. For the initial LSTM layer, the inputs were given as per the data shaped according to the training data with the required number of hidden layers. After being processed by the LSTM cells, for creating the copies (n) equal to the number of defined time steps, a Repeat Vector has been made use of. Now the 1 x (number of time steps) set will be fed to the decoder part of the model, which was created with an LSTM layer along with the Time Distributed layer. Making use of the Keras [10] layer- time distribution was to mainly apply the processed vectors to every slice of the set passed into the decoder. The dimension needed from this time distribution layer was same as the reshaped dataset passed to the model (steps = 1).

When detecting the anomaly using this model, a certain threshold value has been set. It had to find the points that deviate from the majority of the data. When the threshold value is set, it decides how much the observation deviates. Any points going above this are called anomalies. [9] Also, once the threshold was set, the mean absolute error was calculated using the predicted value of the test data against the actual test data. Based on this value, every point with a loss above the threshold value was considered an anomaly.

The mean absolute error and the threshold were calculated as below formula

Mean absolute error loss of test data =

np.mean(np.abs(X_test_pred - X_test), axis= 1)

Threshold value =

np.max(Mean abs error loss of training data) - 0.3

b) ARIMA Model

The ARIMA model is founded on the concept that past values of a time series can be used to predict future values on their own. ARIMA models are very precise and trustworthy if the input data is large and accurate as possible [11]. ARIMA is a combined form of Auto-Regressive (AR) and Moving Average (MA) models along with the differencing added [12]. The python library Statsmodel provides many statistical models, including ARIMA [13] [14]. This model is employed in the current paper after parametrising it using the ‘order’ argument. As it is used as a statistical modelling approach in this paper, instead of predicting anomalies with ARIMA, we have used the below formula to calculate the threshold value to determine the anomalies [15].

Threshold_value = mean(squared_errors) +

*(z * standard_deviation(squared_errors))*

The squared errors of each data point were retrieved from the model after fitting it to the input data with the ‘.resid’ argument. ‘z’ is a constant which can be changed to adjust the threshold value.

c) DBSCAN Model

The DBSCAN clustering model is the simplest method used in this paper for time series anomaly detection. The ability of DBSCAN to detect outliers while clustering was utilized here. Sklearn is a famous open source python library which provides many machine learning models, and the DBSCAN is implemented using this [16]. The parameter called ‘min_samples’ is used to set the number of data points in the cluster. As DBSCAN is robust to the outliers clustering algorithm, the data points which are not part of any clusters can be considered outliers, and they can be categorized as anomalies in the input data.

IV. RESULTS

The observations from the different models that were implemented to find the anomalies from the chosen dataset are as follows:

A. LSTM Model

The model designed using the LSTM and Autoencoder setup, which found the outliers using threshold-based anomaly detection, was trained for ten epochs. Initially, the model tends to have shown signs of overfitting. Then the model was redesigned to overcome this by adding more hidden layers(dropouts), changing the batch size given for the training from 64 to 32 batches and the time steps had to be readjusted for training. The validation loss and the training loss can be seen in Figure 6. Here the training loss shown in the blue line shows the error rate of the model and the validation loss is shown in the orange line.

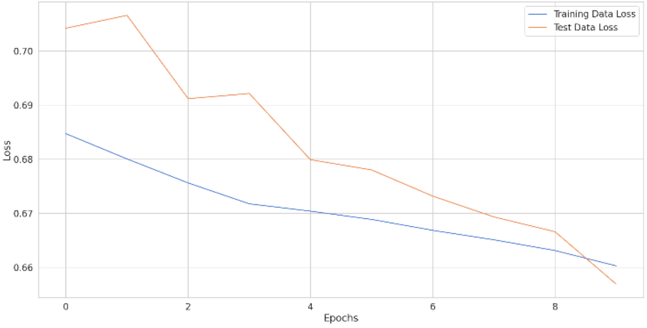

Figure 6 Training Loss and Validation Loss LSTM model

Also, the visualisation of the results obtained by the LSTM model can be seen as shown in Figure 7. Plotted using Python libraries matplotlib [17] and seaborn [18].

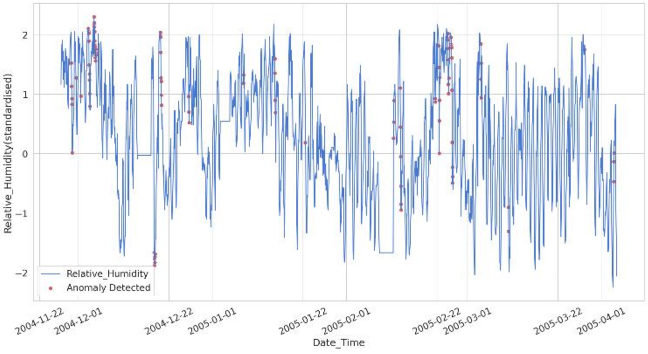

Figure 7 LSTM Model: Anomaly Detection

B. *ARIMA Model*

Once the model was designed based on the order value, it was fitted to the stationary input data consisting of the humidity feature. Then the squared errors of the fitted values were retrieved from the ARIMA model to calculate the threshold to determine the anomalies in the data. The threshold value calculated using the formula mentioned in Section III was about 84.34967. So, all the data points above this threshold were categorised as anomalies and plotted to visualise the results, as shown in Figure 8

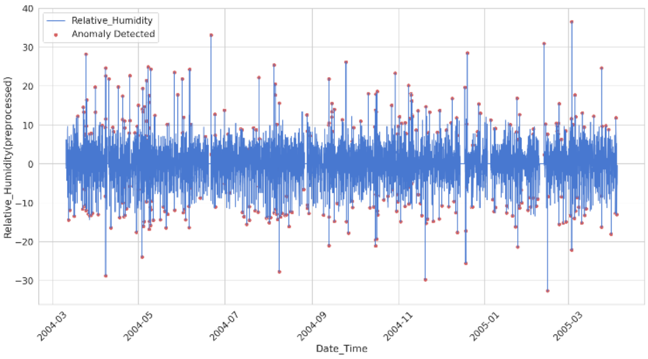

Figure 8 ARIMA model: Anomaly detection

Even though the forecasting of the values was not in the scope of this implementation, we had to forecast the designed model with the steps count of the test data to calculate the mean absolute error of the model. This was done by using the predicted values of the forecasted data and the true value of the test data, which turned out to be about 3.3433.

C. *DBSCAN Model*

The DBSCAN model was developed with a minimum sample of 15. The model has been fit to the input data and collected the labels assigned to each data point by the DBSCAN. The data points with label -1 are not part of any clusters, and we have determined them as anomalies. Then the input data is plotted with these data points highlighted in Figure 9. If the 'min_samples' parameter was changed, the result was also been seem to vary. We have observed that the increase in minimum sample returns more anomalies because the density area is getting short compared to a low minimum sample value.

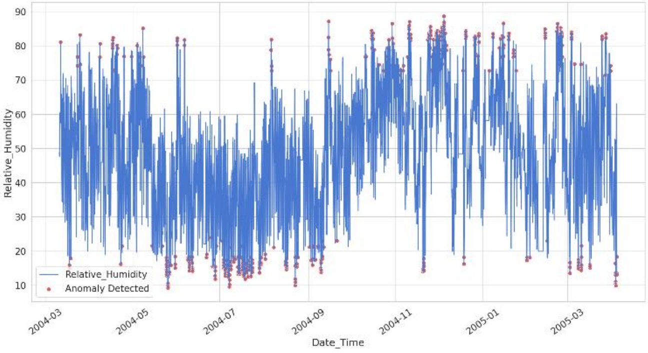

Figure 9 DBSCAN Model: Anomaly Detection

V. DISCUSSION

The data used for time series anomaly detection in this project was sensor data from March 2004 to February 2005 with a frequency of every 1 hour in a day. The dataset had 9357 records, and the relative humidity value was chosen to detect the anomalies as this study implemented univariate anomaly detection [8].

The DBSCAN was found to be the simplest and fastest among the three models chosen in this study as it does not require much preprocessing, and setting the threshold and building the model were simple. Also, observed that increasing the density of the DBSCAN model provides more anomalies, but most of the anomalies found near the high-density area are visible in Figure 9.

In this study, the ARIMA model is used to construct a statistical method for identifying anomalies, and it was found that the technique for determining the threshold had a significant impact on the results. We have also checked the prediction capability of the ARIMA model to compare it with the LSTM-based model, which we have designed and observed that the LSTM was more potent and had a lower error score than ARIMA. While designing the LSTM model, it was seen that, based on the threshold value set, it started to show anomalies. Another point to be noted here is that the time steps passed onto the encoder part of the model can be varied based on what span of the data you need to train.

VI. CONCLUSION

This study has used anomaly detection on time series data and shown how various data points from the huge set of collected data can vary in their pattern, or cycle, from the rest of the data sets/observations. Among the three approaches used in this study, DBSCAN was the fastest and easy way to detect anomalies but noticed that it categorizes many data points as anomalies in a high-density area. LSTM model shows a low error rate compared to ARIMA. It was discovered that the threshold calculation formula used in the statistical method had a significant impact on the outcomes. So, we expect the prediction model will be the safest choice for a reliable time series anomaly detection where it is possible to tune the parameters of the model and evaluate the performance of the model.

VII. References

[1] T.-B. Ottosen and P. Kumar, "Outlier detection and gap filling methodologies for low-cost air quality measurements".

[2] A. BLÁZQUEZ-GARCÍA, J. A. LOZANO, A. CONDE and U. MORI, "A Review on Outlier/Anomaly Detection in Time Series Data".

[3] S. Du, T. Li and S.-J. Horng, "Time Series Forecasting using Sequence-to-Sequence Deep Learning Framework".

[4] A. Bhattacharya, "Towards Data Science," 2020. [Online]. Available: https://towardsdatascience.com/effective-approaches-for-time-series-anomaly-detection-9485b40077f1#:~:text=DBSCAN%20becomes%20the%20most%20obvious,becomes%20very%20easy%20to%20apply.

[5] B. S. Freeman, G. Taylor, B. Gharabaghi and J. Thé, "Forecasting air quality time series using deep learning," 2018.

[6] Z. Chen and Y. F. Li, "Anomaly Detection Based on Enhanced DBScan Algorithm".

[7] S. Wibisono, M. T. Anwar, A. Supriyanto and I. H. A. Amin, "Multivariate weather anomaly detection using DBSCAN clustering algorithm".

[8] "Air Quality Time Series data UCI," [Online]. Available: https://www.kaggle.com/datasets/aayushkandpal/air-quality-time-series-data-uci?resource=download.

[9] Y. Wei, J. Jang-Jaccard, W. Xu, F. Sabrina, S. Camtepe and M. Boulic, "LSTM-Autoencoder based Anomaly Detection for Indoor Air Quality Time Series Data".

[10] "Keras," [Online]. Available: https://keras.io/about/.

[11] I. Priyadarshini, A. Alkhayyat, A. Gehlot and R. Kumar, "Time series analysis and anomaly detection for trustworthy smart homes".

[12] C. Maklin, "Towards DataScience," 2019. [Online]. Available: https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7.

[13] "Statsmodel," [Online]. Available: https://www.statsmodels.org/stable/index.html .

[14] K. Sheridan, T. G. Puranik, E. Mangortey, O. J. Pinon-Fischer, M. Kirby and D. N. Mavris, "An Application of DBSCAN Clustering for Flight Anomaly Detection During the Approach Phase".

[15] "Univariate Time Series Anomaly Detection Using ARIMA Model," [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/univariate-time-series-anomaly-detection-using-arima-model/.

[16] "scikit-learn," [Online]. Available: https://scikit-learn.org/stable/ .

[17] "Matplotlib," [Online]. Available: https://matplotlib.org/.

[18] "Seaborn," [Online]. Available: https://seaborn.pydata.org/generated/seaborn.scatterplot.html.

[19] M. Alizadeh, M. Hamilton, P. Jones and J. Ma, "Vehicle operating state anomaly detection and results virtual reality interpretation".