Volume: 11 Issue: IX Month of publication: September 2023

DOI: https://doi.org/10.22214/ijraset.2023.55763

Anomaly Detection in Time Series Flight Parameter Data Using Machine Learning Approach

Shama Shilpi[1], Shwetank Aryan[2]
[1]Dept. of Mechanical Engineering, Deenbandhu Chhotu Ram University of Science and Technology, Murthal - 131039, India
[2]Dept. of Mining Engineering, Indian Institute of Technology, Kharagpur – 721302, India

Abstract: Infrastructure monitoring is more crucial than ever, especially in the aviation industry. To attempt to overcome the difficulties brought on by the problems given by the exponential growth of connected devices and data volume, this study analyzes the application of machine learning approaches for anomaly identification in time series flight parameter data. The introduction of network telemetry, which automates data collecting, is presented as a remedy, however processing enormous data quantities in real-time still poses a challenge. With a focus on time-series data, the paper explores the role of machine learning in network telemetry anomaly detection. Statistical, proximity-based, deviation-based, and supervised classifiers are used to identify anomalies, or departures from predicted patterns, in flight parameter data. For a few examples of anomalous data, long short-term memory networks (LSTMs) are used. The objective is to provide an effective anomaly detection system that can process complex time series flight data and includes data purification, anomaly discovery, temporal reference, and value prediction. The methodology describes the univariate anomaly detection strategy, in which distinct models record particular patterns for each flight parameter. The findings offer new understanding of thresholds, repeated anomaly correction, and prediction errors. The results show that the method is accurate in separating instances of normal data from those containing anomalies, making it a useful tool for practical applications needing accurate anomaly identification.

I. INTRODUCTION

Currently, it's more important than ever to monitor infrastructure, including networks, systems, and services. It is crucial for a number of reasons, including notifying partial or complete system failure, preventing outages based on the foreseeability of such events, tracking performance, and, last but not least, security detection of system penetration. However, it has become much less clear how to accomplish prompt, dependable, and sound infrastructure monitoring due to the exponential rise in connected devices and traffic volume [1]. It necessitates comprehending the specifics of system operations and being aware of how they affect one another or the entire infrastructure. To more effectively achieve this objective, the idea of network telemetry has been created. It enables the automated, quick, and concurrent collecting of numerous time-series data types from numerous devices. Massive data quantities must be processed, but this is difficult, particularly in terms of timeliness and scalability [2].

Even in extremely large data quantities, machine learning approaches can process, comprehend, and categorize harmful infrastructure behaviors. Despite recent developments in machine learning, its use in network telemetry anomaly detection is still poorly understood and researched. The goal of this work is to provide fresh insight on time-series data anomaly detection [3]. In order to gain valuable insights into how the network and its components are operating, anomaly detection is a crucial part of network and services management. Generally speaking, a generating process produces measurement data. Anomalies result if this generating process behaves strangely as a result of the system's strange behavior or the entity that affects the generating process. By looking at the time-series data produced, aberrant behavior can be detected in its manifestation [4].

Data points that greatly deviate from the majority of the dataset and do not follow the anticipated patterns are referred to as anomalies, also known as outliers. The three main types of anomaly detection techniques are statistical, proximity-based, and deviation-based methods. The fundamental premise behind statistical anomaly detection is that data adheres to a particular distribution model [5]. A data point is considered an anomaly if its likelihood of being produced by this model is less than a set threshold. By taking into account data points that are scattered widely apart from the majority of the typical data points, the proximity-based technique finds anomalies. Reconstruction error is used as an anomaly score in deviation-based anomaly identification.

Using dimension reduction methods like Principal Component Analysis (PCA) or autoencoders, this method reconstructs the data. The reconstruction error is a measurement of the discrepancy between the original data and the rebuilt data. Anomalies are data points with significant reconstruction errors [6, 7].

Classifiers can also be used to detect anomalies when the dataset has enough examples of both the normal and abnormal classes. An abnormal parameter combination can be recognized by a supervised classifier trained on such data. A probability score representing the level of abnormality in novel parameter combinations is provided by this classifier. Utilizing Long Short-Term Memory networks (LSTMs) is advantageous for creating prediction models when there are few examples of the anomalous class. Based on recent and previous timesteps, an LSTM network that has been trained on typical data can predict future values. During testing, differences between the expected and actual numbers can be used to identify probable anomalies.

In their unsupervised operation, autoencoders and their variants learn to reconstitute typical data patterns. The reconstruction error is used as an anomaly score in the deviation-based methodology used for autoencoder anomaly identification. An autoencoder can masterfully reproduce normal patterns with enough practice on normal data. High reconstruction errors because of abnormal data points might be compared to a specified threshold to identify anomalies.

In the present work, the aim is to create a sophisticated anomaly detection system specifically designed for time series data coming from various sensors throughout the course of a flight. This complex dataset is intended to be the input for the proposed system, which is intended to provide a thorough list of identified anomalies together with their accompanying timestamps for each distinct measure. The system also aims to correct anomalous values by replacing them with anticipated values. A revised dataset free of any anomalies is anticipated to be the system's ultimate output. In conclusion, the study aims to tackle the problem of developing an efficient anomaly detection system capable of processing intricate time series data from several sensors throughout a flight. To assure the creation of a trustworthy and accurate dataset, the system's functionalities include anomaly discovery, temporal referencing, value prediction, and data cleansing.

II. LITERATURE REVIEW

There hasn't been a lot of research on anomaly detection despite the recent rise of telemetry in networking. A streaming telemetry-based anomaly identification engine for BGP anomalies was created by the author [8] at Cisco. The clustering algorithm it utilizes, Den- Stream [9], is an outdated one with poor performance. [10] and [11] are two further works that only loosely apply to anomalies and computer networks. The author’s [12] statistical method is used to identify malicious intent and zero-day attacks. According to the study, machine learning methods fail to capture the broader picture of network behavior. To detect fraudulent private exchange phone conversations, the author [13] combine many unsupervised machine learning techniques; nevertheless, their method is limited to off-line data.

Data labeling is a time-consuming and expensive process in anomaly detection situations. It may be subject to bias since it involves human skill. Additionally, labeled datasets containing anomalies frequently have a significant degree of skewness, for example, 95% normal and 5% anomalous. This could have problems, like difficulties with supervised classifier training. Finding or creating instances based on particular abnormalities in civil aviation heavily relies on human skill. Since it is impossible to collect every example of an abnormal flight (an outlier), it is crucial to pick the appropriate examples of labels in order to carry out accurate classification with the fewest labels possible. This is referred to as the label acquisition problem [14-16].

Despite the paucity of directly applicable research in the field of network traffic anomaly detection, there is an increasing tendency to employ machine learning tools for time-series analysis. In order to identify abnormally slow network transfers in real time, the author [17] offer an SVM (Support Vector Machine) [18] based supervised learning algorithm with an emphasis on TCP flows. In order to model and predict sequential data, including time series, the author [19] examine and investigate the HTM (Hierarchical Temporal Memory) [20] artificial neural network. They train the HTM model using offline, unsupervised learning. Sadly, they only cover the forecast aspect of anomaly detection; they do not include the logic used to identify aberrant data.

Multiple Kernel Anomaly Detection (MKAD) created by the author [21] on the basis of [22], which merged discrete and continuous data, is based on kernel functions and a one-class support vector machine. It remains one of the most cutting-edge techniques for finding anomalies in flight data that includes both continuous and discrete data; both studies were a single class problem. Two cluster-based anomaly detection algorithms were created to handle the issue of different classes of flight data: ClusterAD-Flight [23] and ClusterAD- DataSample [24]. These two algorithms not only classified the flights in a binary manner but also determined the flight data norms and spotted any outliers. These algorithms didn't employ exceedance-based standards.

The applicability of deep learning, particularly LSTM [25] and AutoEncoder [26] based algorithms for anomaly detection on time-series data, has been proposed and studied in a few recent publications.

To identify anomalies in industrial large data, the author [24] offer an off-line, variational LSTM learning model based on reconstructed feature representation. The author [27] goal is to forecast fine-grained network traffic in the SDN (Software-Defined Networking) space using an LSTM model. LSTM and SARIMA (Sequential Auto-Regressive Integrated Moving Average) are two supervised machine learning models that the author [28] use to examine various aspects of network traffic forecasting. Sadly, they ignore the rationale behind anomaly detection and solely concentrate on traffic forecast.

The work of [29] was concentrated on the flight's descent phase and used a real-world dataset of a commercial passenger airplane. This work was created by [30] and is based on iOrca [31], which is the scaled or indexed version of Orca [31]. Orca is likewise based on the k nearest neighbor method, much like our work. The approach suggested in this research measures distance between a data point and its neighborhood to compute the density of that data point, as opposed to the Orca, which calculates pairwise distance. The author [32] examined the effectiveness of iOrca and MKAD. Both approaches resulted in an anomalous score. To discriminate between two anomalous data points, anomalies are scored. MKAD classified the entire flight as normal or abnormal, whereas iOrca was able to identify the exact site of the abnormality. While iOrca fared better for single flights, MKAD performed better for groups of flights.

For the purpose of identifying cyberattacks on water distribution systems, the author [33] offer numerous off-line deep learning architectures based on variational AutoEncoders [34]. To improve performance, they compute the Mahalanobis distance [35] rather than the conventional mean square error in the objective function. A deep AutoEncoder model is used in RE-ADTS [36], an unsupervised anomaly detection method that may be used for batch or real-time anomaly identification. On time-series datasets from varied domains, it seems to perform equally well. A dynamic online data mining tool called STAD [37] is an automated framework for identifying cellular network irregularities. It combines several machine learning techniques, including OC-SVM (One-class SVM), SVR (Support Vector Regression), and LSTM.

The algorithm utilized in this paper was similar to that proposed in the work by [38]. The local outlier factor algorithm, on which this study is based, is the source of the local outlier probability (LoOP) technique. The k nearest neighbor's parameter has less of an impact on LoOP. However, a strategy for choosing the ideal value of the parameter k was addressed in this study. The methods used in the study by differ from those used in processing raw data and entering flight data into the algorithm.

The author [35] used a self-organizing map neural network (SOM NN) to detect anomalies in aviation operations. The input layer and output layer are the two layers of this unsupervised neural network approach. The data is represented as a vector in the input layer, and the output layer displays the data in a self-organized fashion. High-dimensional data points are converted into two-dimensional space via SOM NN. The data may be further analyzed thanks to the reduction in dimensionality because the dataset's structure can now be seen more clearly. SOM groups similar data points together as a result, making it simple to spot abnormal data points.

III. METHODOLOGY

The present work's methodology focuses on univariate anomaly identification, in which current model learns the typical patterns connected to specific parameters. Separate models are created for each parameter because they each display unique patterns. Multivariate anomaly detection takes into account data from all sensors, however it has issues with scalability and might result in difficult results to interpret. Furthermore, these methods demand uniform behavior across all matrices, which isn't always possible. The present work use univariate anomaly detection algorithms since the aim is to find anomalies for each parameter on their own.

The dataset being considered consists of a group of time series identified as $Xk = \{xk(1), xk(2), \dots, xk(n)\}$, where index $t = 1, 2, \dots, n$ corresponds to the duration of a flight. Each time series data collection has a kth sensor connected with it that records measurements. Unique flight IDs are used to identify individual flights, and each flight has a unique duration. It's crucial to remember that different sensors have different data recording rates because of differences in the frequency of data collecting. As a result, the data's temporal resolution may vary, and there might be times when more than one data point is recorded in a single second. In conclusion, the dataset is made up of time series data sets (Xk) from various sensors that are indexed by flight duration (t). These sensors have distinct flight IDs that link them to certain flight instances, and they each collect data at a different rate, which could result in different levels of temporal granularity.

A prediction model that can predict one-time step into the future is built as the first stage in the current process. The size of the historical window used to train this model depends on the autocorrelation function (ACF), which spans the current time step and the 50 values before it. Notably, the prediction model is only trained on examples of normal data, allowing it to accurately reflect the properties of normal data patterns. The difference between the expected and actual value for a time step is used as an anomaly score when a new dataset is submitted for prediction. High forecast errors point to possible anomalies.

With the use of this methodology, anomalies can be found starting with the 51st time step. The use of an autoencoder is carried out for the first 50 values. Normal instances from the first 50 values of each parameter are used to train this autoencoder. When faced with a batch of 50 values containing anomalies, the autoencoder does exceptionally well at reconstructing regular patterns, but the reconstruction error for those data points grows. It can accurately identify anomalies by choosing a suitable threshold. Both autoencoders and RNN-LSTM-based prediction models are used in the current methodology. The autoencoder looks for anomalies and substitutes them with newly built values. This clean set is used as input for the RNN-LSTM model to predict the 51st value after the first 50 values are cleansed, with anomalies replaced by their corresponding reconstructed values. This process makes sure that the RNN-LSTM model's predictions are not impacted by the abnormal values from the first 50 observations.

Also taken into consideration is a different tactic. The 50-value lookback makes predictions available starting at X51, therefore the current effort includes an autoencoder to look for anomalies in the first 50 data points. Adding models trained on padded arrays after concatenating an array with 50 iterations of the value X1 is another method. In this instance, the test array is [X1, X1,... (50 times)..., X1, X2,..., Xn]. Predictions are produced for the padded test dataset using models trained on padded training data. We can only set the threshold once using this method. Padding with zeroes could be a way to prevent problems in cases where an anomaly appears at the test array's first member.

IV. RESULT AND DISCUSSION

The following observations were made after running the dataset:

1) It's important to recognize that forecasts made from time steps that have a lot of abnormalities in their past and present values are frequently remarkably untrustworthy. Relying on predictions made in such circumstances increases the likelihood of running into one of two different sorts of errors.
2) With more Autoencoder and RNN-LSTM training, a notable pattern appears. The prediction errors for typical examples constantly tend to become closer to zero as the models go through more training iterations, but anomalous data points consistently show much greater errors. Setting an adequate threshold for anomaly detection is greatly aided by this behavior.
3) A crucial finding is made regarding the accuracy of forecasts made using historical data, particularly when such data include anomalies. Imagine a situation where a sensor consistently produces elevated abnormal results over a long period of time. These unusual numbers are likely to produce similarly high projections. If the sensor produces normal values again after this period of anomalies, the prediction, which is strong and similar to the earlier anomalies, may falsely categorize the normal value as an abnormality. This might cause these false positives to spread across a number of succeeding data points.

Several prediction iterations are used in a strategic strategy to solve this problem. A set threshold is used to conduct an analysis for abnormalities after each forecast cycle. The anomalous value is replaced with the appropriate expected value when this analysis identifies the first anomaly. The prediction procedure is then continued for the ensuing time steps, utilizing the substituted value this time. Anomalies are effectively kept out of the prediction process by using this approach. The total reliability of the anomaly detection system is increased by this iterative procedure, which makes sure that abnormal values do not affect the forecasts.

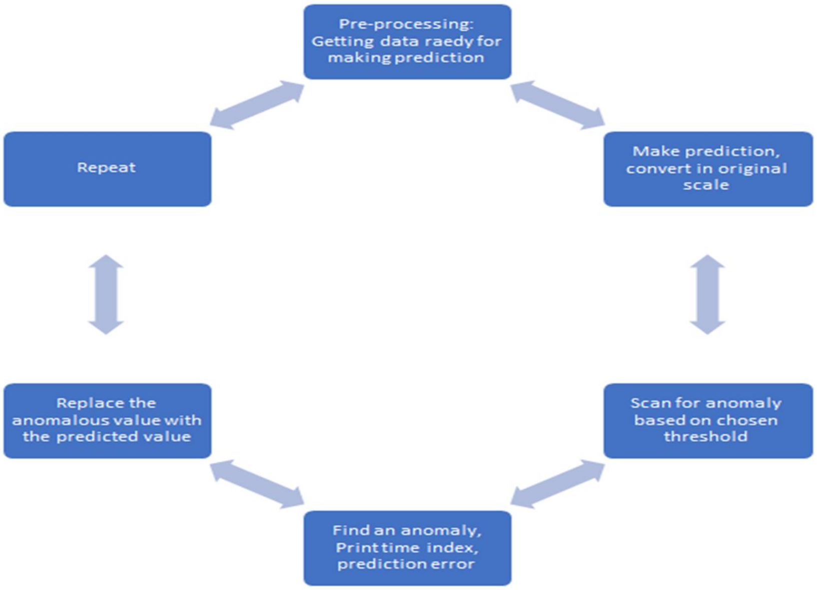
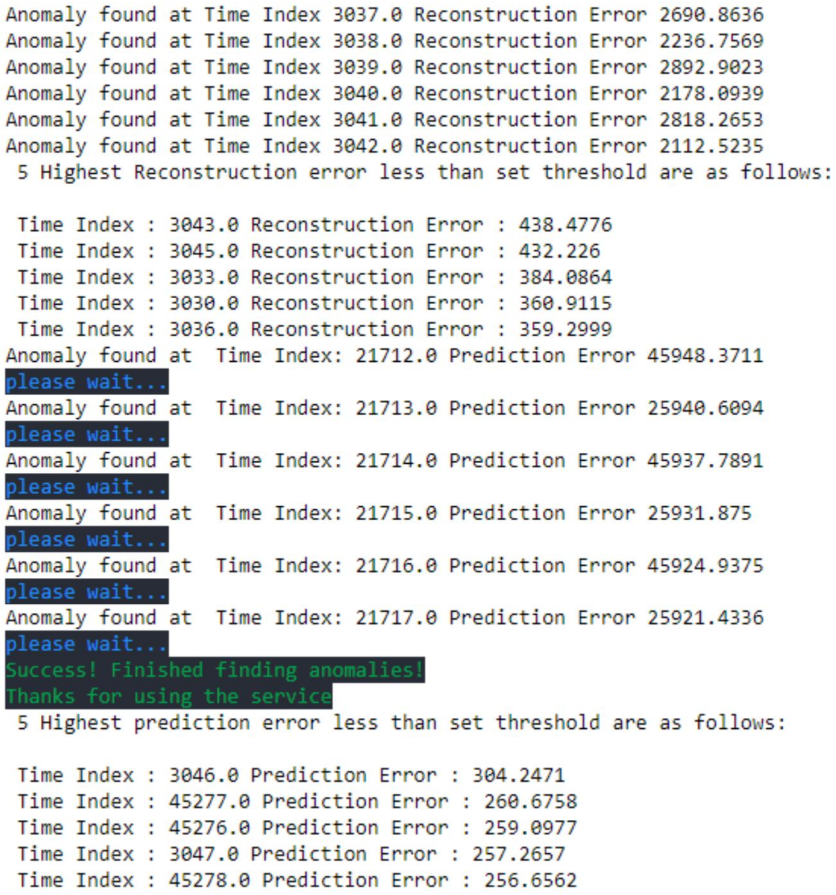

Pre-processing: Getting data raedy for making prediction
Make prediction, convert in original scale
Scan for anomaly based on chosen threshold
Find an anomaly, Print time index, prediction error
Replace the anomalous value with the predicted value
Repeat

Figure 1: Correcting anomalies before using it to make further predictions

When making predictions, models employ past anomalous data if there have been long-term anomalies. The prediction of the model is therefore unreliable. We make predictions and swap out the anomalous value for our anticipated value before the model is used to make more predictions in order to make it work in the case of persistently anomalous data (Fig. 1). The proposed method produces a much better outcome than the conventional strategy of generating a prediction, identifying prediction mistakes, and then making a choice.

Reconstruction errors are first calculated for the first 50 data points. The presence of anomalies is indicated by higher reconstruction errors. The matching time index and its associated reconstruction error are flagged as anomalies if the reconstruction error exceeds a predetermined threshold. The next five highest reconstruction mistakes below the selected threshold are shown in figure 2 after that. Notably, it indicates a need to reevaluate the chosen threshold if the normal point with the highest reconstruction error and the marked anomaly with the lowest reconstruction error are close to each other.



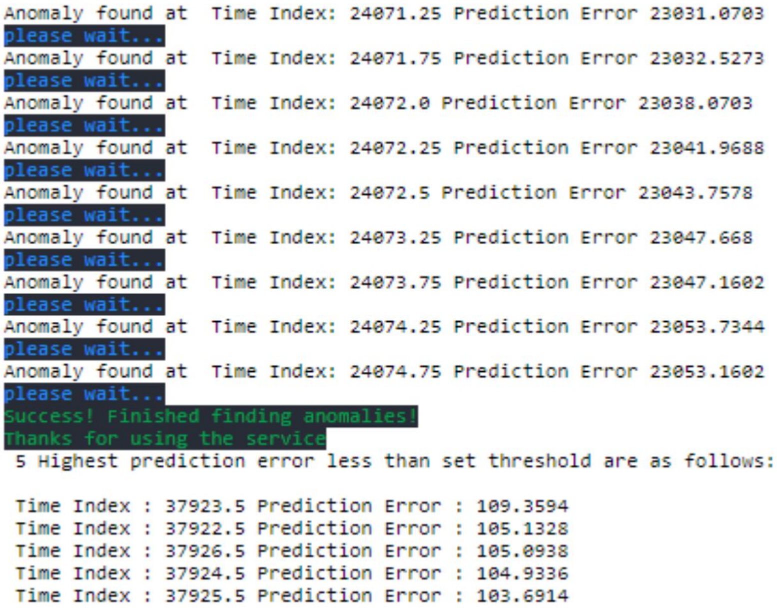
```
Anomaly found at Time Index 3037.0 Reconstruction Error 2690.8636
Anomaly found at Time Index 3038.0 Reconstruction Error 2236.7569
Anomaly found at Time Index 3039.0 Reconstruction Error 2892.9023
Anomaly found at Time Index 3040.0 Reconstruction Error 2178.0939
Anomaly found at Time Index 3041.0 Reconstruction Error 2818.2653
Anomaly found at Time Index 3042.0 Reconstruction Error 2112.5235
 5 Highest Reconstruction error less than set threshold are as follows:

 Time Index : 3043.0 Reconstruction Error : 438.4776
 Time Index : 3045.0 Reconstruction Error : 432.226
 Time Index : 3033.0 Reconstruction Error : 384.0864
 Time Index : 3030.0 Reconstruction Error : 360.9115
 Time Index : 3036.0 Reconstruction Error : 359.2999
Anomaly found at  Time Index: 21712.0 Prediction Error 45948.3711
please wait...
Anomaly found at  Time Index: 21713.0 Prediction Error 25940.6094
please wait...
Anomaly found at  Time Index: 21714.0 Prediction Error 45937.7891
please wait...
Anomaly found at  Time Index: 21715.0 Prediction Error 25931.875
please wait...
Anomaly found at  Time Index: 21716.0 Prediction Error 45924.9375
please wait...
Anomaly found at  Time Index: 21717.0 Prediction Error 25921.4336
please wait...
Success! Finished finding anomalies!
Thanks for using the service
 5 Highest prediction error less than set threshold are as follows:

 Time Index : 3046.0 Prediction Error : 304.2471
 Time Index : 45277.0 Prediction Error : 260.6758
 Time Index : 45276.0 Prediction Error : 259.0977
 Time Index : 3047.0 Prediction Error : 257.2657
 Time Index : 45278.0 Prediction Error : 256.6562
```

Figure 2: Result-RNN-LSTM with Autoencoder

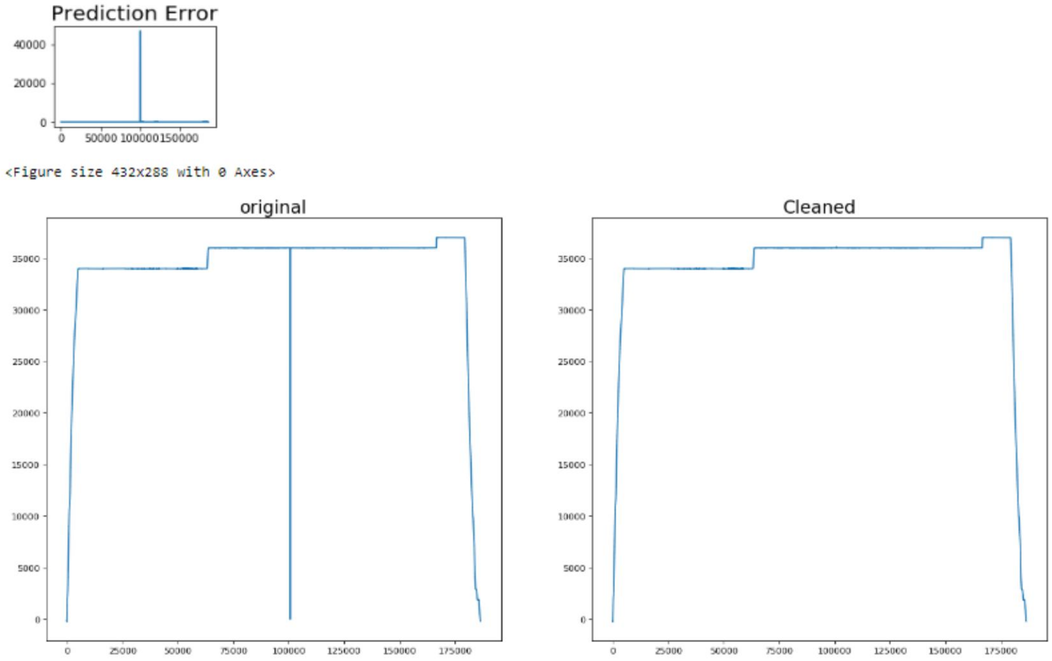
The anomalous values are substituted with the matching reconstructed values once the autoencoder has successfully found anomalies within the first 50 values. From the 51st data point onward, the RNN-LSTM based prediction model is used to forecast using this cleaned initial set of 50 values. Every time an anomaly is found, it is swapped out with the expected value for that particular point. The prediction procedure is then performed using the newly substituted value. Until all anomalous values have been replaced by their matching expected or reconstructed values, this iterative procedure is continued. After this procedure is finished, a properly cleaned dataset is obtained. The RNN-LSTM prediction model is applied once more to this cleaned dataset, and the top five prediction errors are displayed. With the use of the cleaned dataset's prediction errors, the final step serves to discover probable anomalies, offering a thorough evaluation of anomalies that go beyond the initial autoencoder-based detection stage. This method does not necessitate setting a threshold just for RNN-LSTM. Here, an autoencoder is not necessary. We receive predictions for each point in the test array using an RNN-LSTM based prediction model. Without the autoencoder component, the results (Figure 3) are fairly identical to the first method.



```
Anomaly found at  Time Index: 24071.25 Prediction Error 23031.0703
please wait...
Anomaly found at  Time Index: 24071.75 Prediction Error 23032.5273
please wait...
Anomaly found at  Time Index: 24072.0 Prediction Error 23038.0703
please wait...
Anomaly found at  Time Index: 24072.25 Prediction Error 23041.9688
please wait...
Anomaly found at  Time Index: 24072.5 Prediction Error 23043.7578
please wait...
Anomaly found at  Time Index: 24073.25 Prediction Error 23047.668
please wait...
Anomaly found at  Time Index: 24073.75 Prediction Error 23047.1602
please wait...
Anomaly found at  Time Index: 24074.25 Prediction Error 23053.7344
please wait...
Anomaly found at  Time Index: 24074.75 Prediction Error 23053.1602
please wait...
Success! Finished finding anomalies!
Thanks for using the service
 5 Highest prediction error less than set threshold are as follows:

 Time Index : 37923.5 Prediction Error : 109.3594
 Time Index : 37922.5 Prediction Error : 105.1328
 Time Index : 37926.5 Prediction Error : 105.0938
 Time Index : 37924.5 Prediction Error : 104.9336
 Time Index : 37925.5 Prediction Error : 103.6914
```

Figure 3: Result from RNN LSTM model



```
<Figure size 432x288 with 0 Axes>
```

Figure 4: Anomalous signal values & Corrected values

Further threshold selection was performed. Accurate anomaly detection requires a procedure called threshold determination. A thorough investigation of the prediction error distribution was done in order to set an adequate threshold. A subset of altitude-related, well-performing flight data that was thought to be typical of the entire population was used for this investigation. The analysis showed that the distribution of prediction errors follows the pattern of a normal distribution. This distribution's computed mean is 38.0727, and the matching standard deviation is 22.1488. Notably, the highest observed prediction error was 559. Two important techniques are being thought about in the context of threshold selection. In the first method, the threshold is raised above the greatest forecast error found in the dataset of "good" flights. The goal is to prevent any "normal" data point from the successful flights from being mistakenly marked as unusual.

In Fig. 4, left figure represents Anomalous signal values for which the reading was abruptly zero for almost ten seconds, indicating the presence of an anomaly. Left figure represents corrected values for which The predicted outcomes are used to replace the anomalous values, demonstrating that the prediction corresponds with the data's usual trend.

A different option would be to set the threshold at "mean + 3 *" standard deviation. With this approach, abnormalities would be appropriately identified with a remarkable high level of accuracy—roughly 99.14%.

Several methods can be used to further refine the threshold selection process. The histogram of prediction errors can be fitted with a normal distribution curve as one method. This method aids in choosing the best threshold by giving a more comprehensive view of the properties of the distribution. Incorporating more "good" flight cases would result in a more representative population for the study, which would ultimately enhance threshold accuracy while also improving the quality of the analysis.

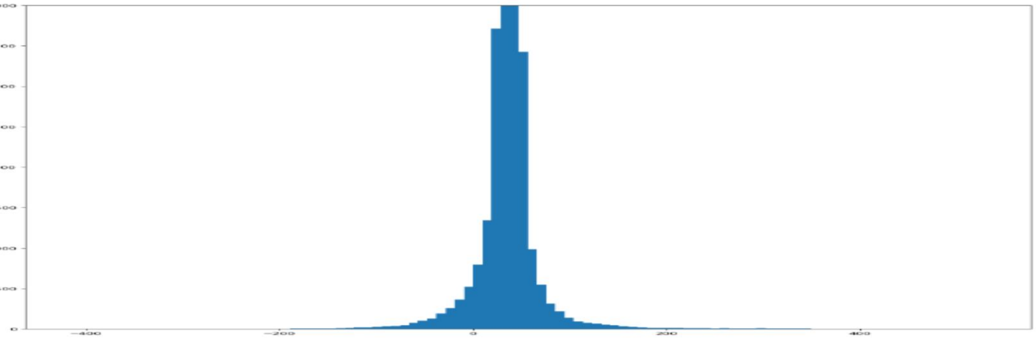


Figure 5: Frequency distribution of prediction errors

Figure 5 depicts the frequency distribution of prediction errors. The determination of the threshold at which a prediction error should be considered an abnormality by analyzing this data can be done. In this distribution, the threshold can be calculated as mean + 3 * standard deviation.

V. CONCLUSION

The current study shows how machine learning approaches can be used in practice for the task of anomaly detection. The present work uses an LSTM-Autoencoder, a model that combines Long Short-Term Memory (LSTM) networks in both the encoder and decoder parts. This is the main component of strategy. The complex patterns connected to typical data instances are successfully captured by the autoencoder architecture while it operates unsupervised. Additionally, the present work make use of RNN-LSTMs networks' predictive abilities, which are trained under supervision to make predictions about the future based on past and present data.

The results of the study have shown encouraging outcomes. It is noteworthy that both the prediction errors produced by the RNN-LSTMs network and the reconstruction errors resulting from the LSTM-Autoencoder model clearly distinguish between normal and anomalous data points. The current work is able to efficiently select a sufficient threshold for anomaly identification due to the distinctness between the errors. By setting this threshold, abnormalities may be reliably found, improving the precision and effectiveness of the present approach. In conclusion, the current study shows how machine learning methods, specifically the LSTM-Autoencoder and RNN-LSTMs networks, can be successfully applied for anomaly detection tasks. The outcomes highlight how well this method distinguishes between typical and abnormal data instances, supporting its potential use in real-world applications requiring precise anomaly identification.

VI. FUTURE SCOPE

The future scope of the present study can be applied to various applications.

1) *Multivariate Anomaly Detection:* Extending our approach to multivariate anomaly detection settings is a potential direction for future research. Through the use of a set of additional parameters as predictors and a supervised learning framework, parameters' values might be predicted using this method. This method increases the accuracy of anomaly detection by using other parameters' predictive capacity in addition to a parameter's own historical values. This approach might offer a more thorough explanation of abnormalities by identifying interdependencies among parameters.

2) *Advanced Time Window Selection:* Choosing the quantity of historical and present values for forecasting is a crucial step in the prediction process. It is essential to thoroughly analyze the Autocorrelation Function (ACF) before making such decisions. It is crucial to reevaluate the use of historical data for prediction if future values show little association with past values. Additionally, examining the parameter's correlations with other parameters provides important information about which parameters should be used as predictors.
3) *Managing Boolean Parameters:* A novel strategy for situations involving Boolean parameter values can involve training a classifier. Based on further pertinent characteristics that affect the parameter, this classifier would forecast Boolean values (0 or 1). Instances where the actual recorded value deviates from the expected value could be reported as potential anomalies by assessing the probability of receiving a given value (for example, 0) and comparing it against a specified threshold. This method improves the accuracy of anomaly detection by making use of the inherent properties of Boolean parameters.

REFERENCES

[1] Srivastava, A.N. Discovering system health anomalies using data mining techniques. In Proceedings of the 2005 Joint Army Navy NASA Airforce Conference on Propulsion, Monterey, CA, USA, 5–8 December 2005.
[2] Das, S.; Matthews, B.L.; Srivastava, A.N.; Oza, N.C. Multiple kernel learning for heterogeneous anomaly detection. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD, Washington, DC, USA, 25–28 July 2010.
[3] Li, L.; Das, S.; John Hansman, R.; Palacios, R.; Srivastava, A. Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations. J. Aerosp. Inf. Syst. 2015, 12, 587–598.
[4] Li, L.; Hansman, R.; Palacios, R.; Welsch, R. Anomaly detection via a Gaussian Mixture Model for flight operation & safety monitoring. Transp. Res. Part C Emerg. Technol. 2015, 64, 45–57.
[5] Melnyk, I.; Banerjee, A.; Matthews, B.; Oza, N. Semi-Markov switching vector autoregressive model-based anomaly detection in aviation systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
[6] Das, S.; Sarkar, S.; Ray, A.; Srivastava, A.; Simon, D. Anomaly detection in flight recorder data: A dynamic data-driven approach. In Proceedings of the 2013 American Control Conference, Washington, DC, USA, 17–19 June 2013.
[7] Bhaduri, K.; Matthews, B.L.; Giannella, C.R. Algorithms for speeding up distance-based outlier detection. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD, San Diego, CA, USA, August 2011.
[8] Bay, S.; Schwabacher, M. Mining Distance-Based Outliers in Near Linear Time with Randomization and A Simple Pruning Rule. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD, Washington, DC, USA, 24–27 August 2003.
[9] Matthews, B.; Das, S.; Bhaduri, K.; Das, K.; Martin, R.; Oza, N. Discovering Anomalous Aviation Safety Events Using Scalable Data Mining Algorithms. J. Aerosp. Inf. Syst. 2014, 11, 482.
[10] Oehling, J.; Barry, D. Using Machine Learning Methods in Airline Flight Data Monitoring To Generate New Operational Safety Knowledge From Existing Data. Saf. Sci. 2019, 114, 89–104.
[11] Megatroika, A.; Galinium, M.; Mahendra, A.; Ruseno, N. Aircraft anomaly detection using algorithmic model and data model trained on FOQA data. In Proceedings of the 2015 International Conference on Data and Software Engineering (Icodse), Yogyakarta, Indonesia, 25–26 November 2015.
[12] J. Hawkins, S. Ahmad, S. Purdy, and A. Lavin, “Biological and Machine Intelligence (BAMI),” Initial online release 0.4, 2016.
[13] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” AIChE Journal, vol. 37, no. 2, pp. 233–243, 1991.
[14] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, “Variational LSTM Enhanced Anomaly Detection for Industrial Big Data,” IEEE Trans- actions on Industrial Informatics, vol. 17, no. 5, pp. 3469–3477, 2021.
[15] A. Lazaris and V. K. Prasanna, “An LSTM framework for modeling network traffic,” in Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, 2019, pp. 19–24.
[16] J. Soheil et al., “On the Practicality of Learning Models for Network Telemetry,” Proceedings of Network Traffic Measurement and Analysis Conference.
[17] G. Box and G. Jenkins, Time Series Analysis: Forecasting and Controls rev. Oakland California: Holden-Day, 1976.
[18] L. Gjorgiev and S. Gievska, “Time Series Anomaly Detection with Variational Autoencoder Using Mahalanobis Distance,” in Proceedings of the ICT Innovations 2020. Machine Learning and Applications, V. Dimitrova and I. Dimitrovski, Eds., Cham: Springer International Publishing, 2020, pp. 42–55.
[19] P. Diederik and M. Welling, Auto-Encoding Variational Bayes, 2013. arXiv:1312.6114.
[20] G. McLachlan, “Mahalanobis distance,” Resonance, vol. 4, no. 6, pp. 20–26, 1999.
[21] T. Amarbayasgalan, V. H. Pham, N. Theera-Umpon, and K. H. Ryu, “Unsupervised Anomaly Detection Approach for Time-Series in Multi-Domains Using Deep Reconstruction Error,” Symmetry, vol. 12, no. 8, 2020.
[22] A. Dridi, C. Boucetta, S. E. Hammami, H. Afifi, and H. Moungla, “STAD: Spatio-Temporal Anomaly Detection Mechanism for Mobile Network Management,” IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 894–906, 2021.
[23] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J.
[24] C. Platt, “Support Vector Method for Novelty Detection,” Advances in Neural Information Processing Systems, pp. 582–588, 2000.
[25] H. Drucker, C. C. Burges, L. Kaufman, A. J. Smola, and V. N. Vapnik, “Support Vector Regression Machines,” in Proceedings of the Advances in Neural Information Processing Systems, ser. NIPS 1996, vol. 4, MIT Press, 1997, pp. 155–161.
[26] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, “Network Anomaly Detection Using LSTM Based Autoencoder,” in Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, ser. Q2SWinet ’20, Alicante, Spain: Association for Computing Machinery, 2020, pp. 37–45.
[27] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença, “Long Short-Term Memory and Fuzzy Logic for Anomaly Detection and Mitigation in Software-Defined Network Environment,” IEEE Access, vol. 8, pp. 83 765–83 781, 2020.

[28] T. J. Lee, J. Gottschlich, N. Tatbul, E. Metcalf, and S. Zdonik, Greenhouse: A Zero-Positive Machine Learning System for Time- Series Anomaly Detection, 2018.

[29] J. Hochenbaum, O. S. Vallis, and A. Kejariwal, Automatic Anomaly Detection in the Cloud Via Statistical Learning, 2017. arXiv:1704.07706 [cs.LG].

[30] M.-C. Lee, J.-C. Lin, and E. G. Gran, How Far Should We Look Back to Achieve Effective Real-Time Time-Series Anomaly Detection? 2021. arXiv:2102.06560 [cs.LG].

[31] A. Lavin and S. Ahmad, "Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark," in Proceedings of the IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015, pp. 38–44.

[32] I. Numenta. (). "NAB: Numenta Anomaly Benchmark [Online code repository]," [Online]. Available: https://github.com/numenta/NAB. Accessed: 2021-01-13.

[33] H. Ren et al., "Time-series anomaly detection service at microsoft," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ser. KDD '19, Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 3009–3017.

[34] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification," IEEE Access, vol. 6, pp. 1662–1669, 2018.

[35] Breunig, M.; Kriegel, H.; Ng, R.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data—SIGMOD, Dallas, TX, USA, 15–18 May 2000.

[36] Boeing. Statistical Summary of Commercial Jet Airplane Accidents; Boeing: Seattle, WA, USA, 2021; p. 14.

[37] Airbus. A Statistical Analysis of Commercial Aviation Accidents 1958–2021; Airbus: Blagnac, France, 2022; p. 27.

[38] Aggarwal, C.; Hinneburg, A.; Keim, D. On The Surprising Behavior of Distance Metrics in High Dimensional Space. In Database Theory—ICDT; Van den Bussche, J., Vianu, V., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 1973, pp. 420–434.