IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Analysis Anomalies And False Positive Rate In A Network Traffic Data

[1]P Vaishnavi, [2] G. Ramanjinamma, [3]Poornima Gowda H S, [4]Spandana S M, [5]Varshitha N, [6]Nikita
[1]Student, [2]Professor, [3]Professor, [4]Student, [5]Student, [6]Student
[1]Computer Science and Engineering,
[1]Sai Vidya Institute of Technology, Bangalore, India

Abstract: This article gives an overview about how the anomalies are detected in the network data containing the bytes sent and received and session duration while a bit of data is being transferred. The anomalies in the data play an important role in detecting security breaches, performance issues and abnormal behaviours while the network is being transmitted.

The identification of anomalies face challenges, particularly when it comes in finding and managing false positives i.e., incorrectly flagged activities as abnormal or malicious. This paper aims in detecting all the anomalies in the given data and finding false positives in the data set. This paper gives different views about the different methods to detect anomalies in the given network data based on the false positive rates.

We discuss different algorithms that are used in this detection including machine-learning algorithms. Here we also discuss the causes, and additionally we display the graph of the anomalies in the data. The analysis is used for improving the effectiveness of Intrusion Detection Systems for future.

Keywords- False Positive, True negatives

1.INTRODUCTION

In the era of growing cyberthreats and attacks, network traffic data analysis is an essential component of computer networks. The primary goal of the study is to identify anomalies, which are simply departures from normal behaviour or any malicious activity such as intrusions, malware, infections, or network breaches. Anomalies include things like strange data flow patterns, unanticipated resource usage throughout the data flow, or poor connectivity between two or more devices. Early detection of these irregularities will help us avoid many security breaches and ensure seamless network transfer or communication between two or more devices.

A false positive occurs when normal or innocuous networks are reported to contain anomalous or dangerous data. Overwhelming false positive rates or an increase in false positive rates could make network managers' jobs more difficult overall and possibly lead them to ignore actual threats due to alarms. Finding anomalies with a low false positive rate is a crucial feature of network monitoring systems and the main requirement for reducing false positive rates.

Attacks are become increasingly complex due to the diverse nature of network trafficking, which makes it harder to distinguish between legitimate and dubious activities in data sets. Statistical models, behaviour analysis, machine learning algorithms, and deep learning algorithms are some of the techniques used to find anomalies; nevertheless, each has limitations.The main difficulty with anomaly detection systems is differentiating between sensitivity and specificity.

1.1 Dependency of False Positive rates

False positive rate (FPR) is one indicator used to evaluate binary classification systems. It is defined as the proportion of true negative cases, or examples that are normal or non-anomalous, that are incorrectly classified as positive cases, or instances that are unusual or problematic.

It represents the likelihood that a normal event could be incorrectly labelled as an anomaly. This is how the False Positive Rate is calculated: True Negatives plus False Positives/False Positives

False positives are the number of normal cases that were incorrectly labelled as abnormalities.

True Negatives (TN) are the number of normal cases that were correctly identified as normal.

A model with a false positive rate around zero has a high specificity, which means it generates few false alarms; on the other hand, a high FPR means the model often misclassifies typical behaviour as abnormal, which can be problematic in real-world applications.

1.2 Data type and Input Structure

We provide a data file for the system that has columns with the following values: timestamp, source_ip, destination_ip, bytes_sent, bytes_received, session_duration, label, and ground_truth.

The timestamp contains the date and time of the data's transmission.

The IP address of the source from which the data was sent is known as source_ip.

The IP address of the destination, where the data must be sent, is known as destination_ip.

Bytes_sent: At a specific moment in time, the quantity of bytes transmitted to the destination address.

Bytes_received: The quantity of bytes that the destination address has received at a specific moment in time.

Session_duration: How long does it take to move data or from the originating IP address to the destination address?

Ground_truth values: These values serve as a baseline for identifying irregularities.

2.METHODOLOGY OF ANOMALY DETECTION

In this article we describe how to detect anomalies in the network traffic and here in this section we describe about the method we use to implement this. The analysis of anomalies and false positives in network traffic data is a critical component of network security. It involves a structured process of collecting, preprocessing, analyzing, and interpreting data to detect malicious activities while minimizing false alarms. A well-defined methodology is essential to ensure accurate detection of security breaches and to avoid overwhelming security teams with unnecessary alerts.

2.1 System Design

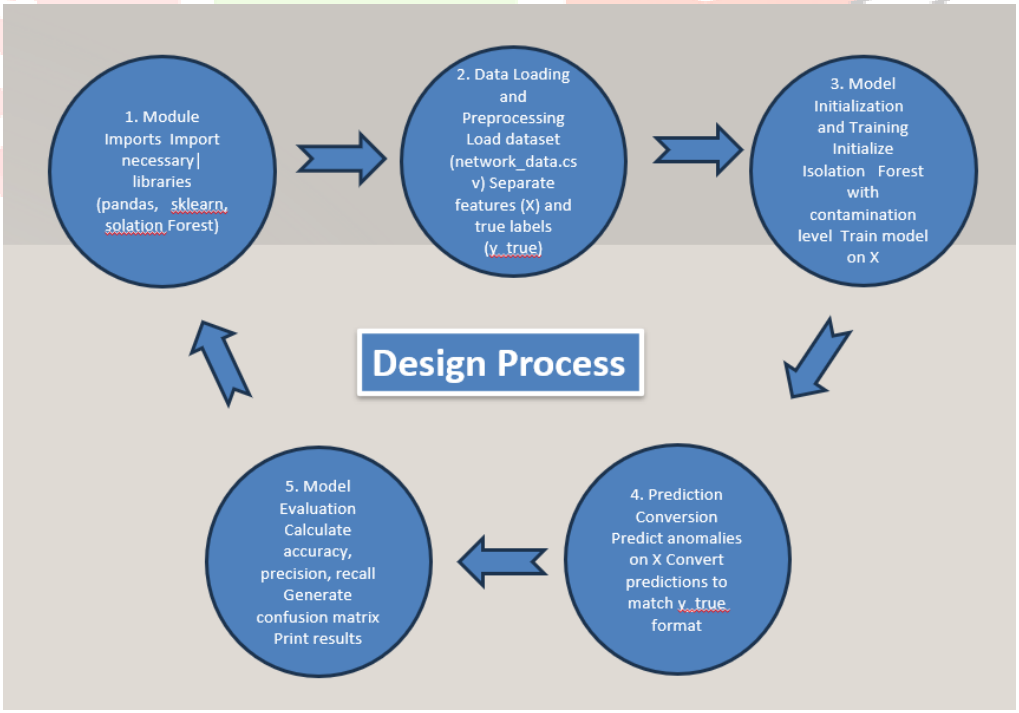

Figure 1: System design process for anomaly detection

This project begins with data collection and preprocessing. Choosing a relevant dataset that encompasses both normal and anomalous network activity is crucial. NSL-KDD, which provides tagged network traffic data, and UNSW-NB15, another dataset with a range of attack types, are excellent choices. The dataset should have attributes such as timestamp, source_ip, destination_ip, bytes_sent, bytes_received, session_duration, and labels. Numerical features such as bytes_sent, bytes_received, and session_duration are scaled or normalised during preprocessing. Additionally, IP addresses need to be converted into a format that may be used, such as one-hot encoding or IP cluster representation.

Model selection and anomaly detection come next. Using an unsupervised anomaly detection technique, like Local Outlier Factor (LOF), One-Class SVM, or Isolation Forest, is a good place to start. Features like bytes_sent, bytes_received, and session_duration will be used as inputs in these models. Normal data points are used for training in order to identify common patterns, and a contamination rate is established depending on the dataset or existing knowledge. Testing with labelled data, if available, is part of the initial model evaluation process in order to obtain a baseline measurement of false positives. Measures such as false positive rate, recall, accuracy, and precision are noted.

Reducing false positives is an important first step. This means looking at false positives to identify patterns, such as certain IP addresses, data amounts, or session durations that are frequently reported as anomalous without justification. Model changes or feature engineering are informed by acquired knowledge. By including context, improved feature engineering can differentiate between daytime and nighttime traffic patterns, generate role-based user profiles, and organise IPs by subnet or region to differentiate between internal and external traffic. Changing the contamination parameter, playing with model topologies and parameters, and trying out other methods like an ensemble of LOF and Isolation Forest can all help with cross-validating anomalies. Using techniques like Random Forests or Gradient Boosting, supervised models can be tested if labels are supplied.

Monitoring metrics and assessing the model are crucial for this project. Important measures include false positive rate, recall, accuracy, and precision. Adjusting the threshold for anomaly scores can help reduce false positives while maintaining true anomaly detection. Visualisation and reporting are essential for understanding and evaluating the model's performance. Anomaly timelines, confusion matrices, and ROC curves (for supervised algorithms) are examples of visualisations that provide insight into patterns and trade-offs between true positive and false positive rates.

Finally, summarising findings is crucial for comprehensive reporting. This means explaining how feature engineering and model tuning affect false positives, giving details on the types of traffic or sessions that cause false positives, and highlighting issues and potential improvements for real-world applications. Future studies could focus on improving these aspects to create an anomaly detection system that is more accurate and dependable.

3. IMPLEMENTATION USING LOCAL OUTLIER FACTOR ALGORITHM

This Flask web application is designed to detect anomalies in uploaded Excel files using machine learning. It begins by importing necessary libraries such as Flask for web framework, pandas for data manipulation, matplotlib for plotting, and Local Outlier Factor from scikit-learn for anomaly detection. The Flask instance is created, and routes are defined for the home page, file upload, and plotting functionality. When a user uploads a file, the 'upload_file' route handles the request, reads the Excel file into a DataFrame, and calls the 'detect_anomalies' function. This function trains a Local Outlier Factor model on the numerical columns of the DataFrame to predict anomalies. Anomalies and false positives are identified, and the false positive rate is calculated. The results, including a plot of the data with anomalies highlighted, are rendered in an HTML template. The application runs in debug mode to provide detailed error messages, ensuring easy troubleshooting during development.

To design our web page we import Flask, request, render_template,send_file from the flask library and we import few more libraries such as pandas,matplotlib and few more algorithms such as LocalOutlierFactor and io library

The main usage of the flask library is to handle the web requests and responses.Pandas is a powerful data manipulation and anlysis library.matplotlib is imported for creating static, animated and interactive visualizations.We import LocalOutlierFactor module from the scikit-learn library which is mainly to detect anomalies in the dataset.We import the io module for handling all the input and output functions in the program.We create an instance of the Flask class for our web application.Later we define a route for the root URL ('/').Then we define index function to handle requests to root the URL. This function renders the 'index.html' template when the root URL is accessed.

We define a route to upload the file which accepts only POST requests.Define the upload_file function to handle file uploads.A function should be designed to retrieve the uploaded file from the request.We describe a variable to read the file and upload excel file into a pandas DataFrame.The function “anomalies, plot_url, fp_rate = detect_anomalies(df)” which calls detect_anomalies function to detect anomalies in DataFrame and retrieve the plot URL and false positive rate. We define the function which renders the 'results.html' template with the anomalies table, plot URL, and false positive rate.

4. EXPECTED OUTCOMES

Achieving a reduced false positive rate is crucial for the effectiveness of any anomaly detection system. By targeting a specific rate, such as less than 5%, through the application of improved algorithms and the incorporation of user feedback loops, the system can significantly enhance its reliability. Alongside this, improving detection accuracy involves increasing the overall accuracy of the anomaly detection process. This means ensuring metrics show an increase in true positive rates while maintaining low false positives, thus boosting the system's trustworthiness.

User satisfaction is another critical aspect, as obtaining positive feedback regarding the relevance and quality of alerts can lead to reduced alert fatigue. This can be reflected in user surveys and overall user experience. Providing actionable insights from detected anomalies is essential, as it enables users to respond effectively and timely to potential issues, ensuring that the detected anomalies translate into meaningful actions.

Comprehensive reporting is also vital, as it generates detailed reports summarizing detection performance. These reports should include metrics on true positives, false positives, missed detections, and user feedback, providing a clear picture of the system's effectiveness. Efficient resource utilization is another goal, ensuring the system's performance and resource usage are optimized to minimize computational load while processing incoming data streams.

Continuous improvement is facilitated by establishing a feedback loop where user feedback leads to iterative enhancements in the anomaly detection models, thus improving system performance over time. Seamless integration with existing systems and workflows allows for easy data sharing and collaboration between teams, enhancing the system's utility in a broader operational context.

Increased detection speed is also an objective, aiming to improve the speed of anomaly detection processes. Defined metrics should show reduced latency in alert generation, ideally with alerts generated within one second of data ingestion. Lastly, ensuring compliance and security assurance is crucial. The system should meet regulatory compliance and security standards, resulting in successful audits and a reduced risk of data breaches, thereby maintaining the integrity and security of the data being processed.

5.Result

The result of this detection is that it displays all the rows which contains anomalies and plots the graph of the all the data and displays it using the matplotlib module. Then it also calculates false positive rate of the given dataset. These rates are taken by the whole dataset and calculated using the formula and the algorithm used to detect anomalies is local outlier factor.

There are different algorithms such as One-Class SVM,Autoencoders in neural networks, DBSCAN (Density-Based Spatial Clustering of Applications with Noise),Elliptic envelope and different variations of Isolation forest.These algorithms have few advantages and disadvantages also.We choose the best efficient algorithm among them that is Local Outlier Factor algorithm has better efficiency in finding out the anomalies in the given dataset.

Anomaly Detection Results

	timestamp	source_ip	destination_ip	bytes_sent	bytes_received	session_duration	label	ground_truth	predicted_anomaly
12	2024-12-01 11:56:35	192.168.160.140	10.51.221	1712	159	99.22	0	0.17	-1
14	2024-12-01 11:56:51	192.168.21.19	10.79.152	5631	437	44.16	0	0.15	-1
16	2024-12-01 11:56:56	192.168.92.69	10.199.233	50	1795	64.46	0	0.45	-1
18	2024-12-01 11:57:02	192.168.168.252	10.169.254	7993	7896	35.94	0	0.45	-1
26	2024-12-01 11:57:50	192.168.38.36	10.79.51	5947	1322	35.03	0	0.17	-1
28	2024-12-01 11:58:10	192.168.119.32	10.132.73	5376	1013	34.78	0	0.26	-1
32	2024-12-01 11:58:31	192.168.197.15	10.174.89	5707	1152	25.35	0	0.29	-1
37	2024-12-01 11:58:58	192.168.79.177	10.84.255	332	6351	44.92	1	0.83	-1
64	2024-12-01 12:01:12	192.168.51.18	10.115.238	6172	1814	23.68	0	0.11	-1
65	2024-12-01 12:01:18	192.168.10.121	10.128.226	2805	7974	85.11	0	0.36	-1
71	2024-12-01 12:01:50	192.168.93.30	10.61.166	1356	88	71.45	0	0.25	-1
107	2024-12-01 12:04:42	192.168.141.243	10.127.5	921	7856	70.69	0	0.05	-1
130	2024-12-01 12:07:07	192.168.153.84	10.37.102	7740	2446	29.52	0	0.07	-1
131	2024-12-01 12:07:12	192.168.111.77	10.221.33	755	7557	7.81	0	0.23	-1
158	2024-12-01 12:09:38	192.168.219.82	10.174.118	7989	4920	55.22	1	0.91	-1
160	2024-12-01 12:09:55	192.168.191.228	10.209.11	530	7277	16.65	0	0.25	-1
174	2024-12-01 12:11:17	192.168.202.73	10.45.197	6242	2178	97.76	0	0.07	-1
201	2024-12-01 12:13:50	192.168.29.227	10.206.186	24	1598	57.69	0	0.11	-1
207	2024-12-01 12:14:27	192.168.237.130	10.177.94	2039	7769	55.71	0	0.13	-1
219	2024-12-01 12:15:38	192.168.90.179	10.249.136	5103	131	99.06	0	0.27	-1
233	2024-12-01 12:16:56	192.168.247.128	10.172.67	716	7689	69.52	0	0.21	-1
251	2024-12-01 12:18:36	192.168.22.213	10.173.223	7995	7480	54.28	0	0.33	-1
253	2024-12-01 12:18:50	192.168.243.12	10.236.69	177	4629	10.14	0	0.44	-1
274	2024-12-01 12:20:52	192.168.43.184	10.101.155	402	7938	37.34	0	0.08	-1
281	2024-12-01 12:21:23	192.168.251.229	10.38.160	2615	7602	13.79	0	0.08	-1
285	2024-12-01 12:21:59	192.168.125.38	10.35.129	4402	7815	95.41	0	0.35	-1

Figure 2: Result of the detection

989	2024-12-01 13:26:51	192.168.159.91	10.208.97	5445	680	57.62	0	0.50	-1
992	2024-12-01 13:27:12	192.168.110.133	10.82.37	131	6618	89.19	0	0.02	-1

False Positive Rate

0.14285714285714285

Figure 3: False Positive Rate in the given dataset

Graph of Anomalies

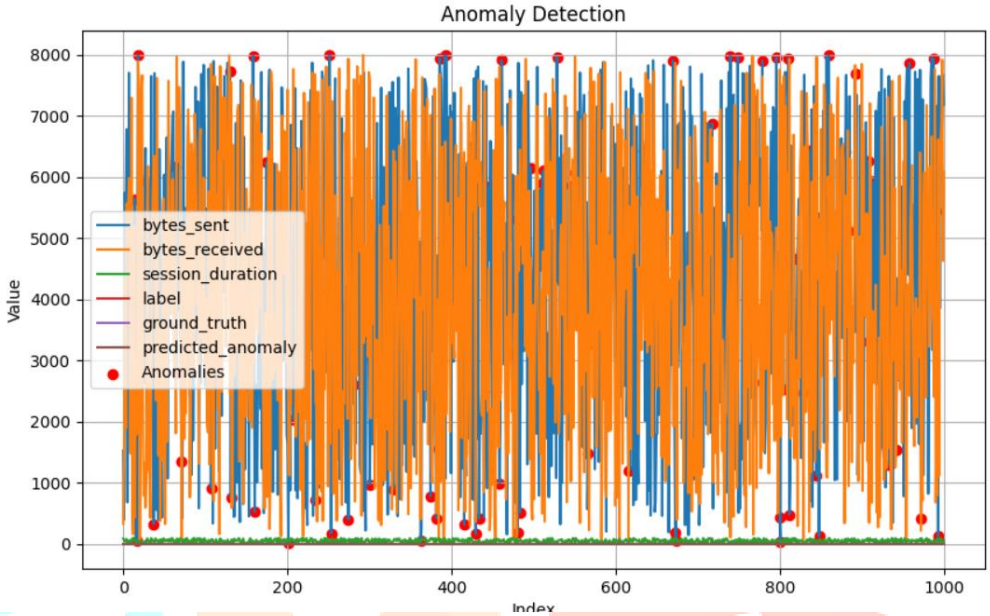

Figure 4: Graph of the anomaly detection

6.CONCLUSION

In summary, developing an anomaly detection system focused on minimizing false positives is a critical endeavor that can significantly enhance decision-making processes across various domains. By clearly defining functional and non-functional requirements, you can establish a robust framework that not only detects anomalies effectively but also ensures user satisfaction and operational efficiency.

The expected outcomes of reduced false positive rates, improved accuracy, actionable insights, and continuous improvement will lead to a more reliable and user-friendly system. Furthermore, the emphasis on scalability, performance, and compliance ensures that the solution can adapt to evolving needs while maintaining data integrity and security.

Ultimately, the successful implementation of this system will empower organizations to proactively identify and respond to anomalies, thus mitigating risks and optimizing operations. As technology continues to advance, the commitment to refining anomaly detection capabilities will be essential in navigating complex data environments and making informed decisions.

This project successfully demonstrates an effective methodology for real-time anomaly detection in cybersecurity, with a strong focus on minimizing false positives. By employing the Isolation Forest model, we achieved a balance between detecting true network anomalies and reducing unnecessary alerts, enabling security analysts to focus on legitimate threats. The integration of a feedback loop further enhances model performance over time, adapting to evolving network patterns and threat landscapes. With this system, organizations can improve their security posture, proactively addressing anomalies and safeguarding against potential intrusions.

REFERENCES

1. Mijalkovic, J., & Spognardi, A. (2022). Reducing the False Negative Rate in Deep Learning Based Network Intrusion Detection Systems. *Algorithms*, *15*(8), 258. https://doi.org/10.3390/a15080258
2. Forstmeier, W., Wagenmakers, E. J., & Parker, T. H. (2017). Detecting and avoiding likely false-positive findings–a practical guide. *Biological Reviews*, *92*(4), 1941-1968.
3. Donchev, D., Vassilev, V., Tonchev, D. (2021). Impact of False Positives and False Negatives on Security Risks in Transactions Under Threat. In: Fischer-Hübner, S., Lambrinoudakis, C., Kotsis, G., Tjoa, A.M., Khalil, I. (eds) Trust, Privacy and Security in Digital Business. TrustBus 2021. Lecture Notes in Computer Science(), vol 12927. Springer, Cham. https://doi.org/10.1007/978-3-030-86586-3_4
4. Maggi, F., Matteucci, M., & Zanero, S. (2009). Reducing false positives in anomaly detectors through fuzzy alert aggregation. *Information Fusion*, *10*(4), 300-311.
5. Jeffrey, N., Tan, Q., & Villar, J. R. (2023). A review of anomaly detection strategies to detect threats to cyber-physical systems. *Electronics*, *12*(15), 3283.
6. Shaik, A. S., & Shaik, A. (2024, April). AI Enhanced Cyber Security Methods for Anomaly Detection. In *International Conference on Machine Intelligence, Tools, and Applications* (pp. 348-359). Cham: Springer Nature Switzerland.
7. Al Jallad, K., Aljnidi, M., & Desouki, M. S. (2020). Anomaly detection optimization using big data and deep learning to reduce false-positive. *Journal of Big Data*, *7*, 1-12.
8. Rizvi, M. (2023). Enhancing cybersecurity: The power of artificial intelligence in threat detection and prevention. *International Journal of Advanced Engineering Research and Science*, *10*(05).
9. Jeffrey, N., Tan, Q., & Villar, J. R. (2023). A review of anomaly detection strategies to detect threats to cyber-physical systems. *Electronics*, *12*(15), 3283.
10. Folino, G., Otranto Godano, C., & Pisani, F. S. (2023). An ensemble-based framework for user behaviour anomaly detection and classification for cybersecurity. *The Journal of Supercomputing*, *79*(11), 11660-11683.
11. Ji, I. H., Lee, J. H., Kang, M. J., Park, W. J., Jeon, S. H., & Seo, J. T. (2024). Artificial intelligence-based anomaly detection technology over encrypted traffic: a systematic literature review. *Sensors*, *24*(3), 898.