# Machine Learning-Driven Anomaly Detection: Strengthening Siem Tools For Robust Cyber Defense

**Rohit Arora (M.Tech.), Vikash Kumar Kharbas (Asst. Prof.)**

*Department of Computer Science & Engineering*

*Vivekananda Global University, Jaipur, Rajasthan (303012)*

***Abstract*****:** In the present growing era of technology organizations are producing a large amount of data every day. The security of that data is mainly based on the proper monitoring and prevention technologies. Threat prevention and monitoring in IoT device logs are very important. However, monitoring individually each device is not an easy task. SIEM security information technology is a platform to overcome this issue by collecting all logs in a centralized platform. But traditional SIEM tools also have detection-based issues, they perform only rule-based detection and cannot detect advanced threats and generate many false positives, because of this IT professionals cannot focus on the real threats. To overcome this concern Machine learning is the approach that can detect advanced threats by behaviour-based analysis as per past data. in this research, we used Isolation Forest Algorithm which is best to separate the normal instances and anomalies. This Machine learning approach reduces the false positive rates and increases anomaly detection. this paper aims to reveal the possible changes in the cybersecurity sphere due to the implementation of machine learning and promote further development of the technology as an addition to the existing SIEM systems. Adopting these advancements is an effective way of strengthening the security of any organization, thus providing a safer and more secure digital environment.

***Keywords*****:** SIEM, False Positives, Machine Learning, Isolation Forest Algorithm

## I. Introduction

In the current trend of captivating virtual structured environments, firms from different fields create a huge amount of log data on a daily basis. These logs are useful for assessing how different systems are performing, managing the workings of the system and more importantly the security of the system (Cao, Qiao and Lyu, 2017). Console logs are at the frontline of detecting and preventing several types of security breaches, thus being a part of solid cybersecurity measures. Conventional SIEM solutions have been used to aggregate logs, assist with real-time monitoring, and allow for the automation of the analysis of security incidents (He et al., 2021). However, the usage of SIEM tools has several inherent problems mainly associated with the enhanced velocity and variety of logs. They frequently use only correlation rules and signatures on which the system is configured, which results in high numbers of false positives and missed threats. Thus, the imperfection and inapplicability of traditional SIEM tools become noticeable as cyber threats are changing and becoming more complicated (Yadav, Kumar and Dhavale, 2020).

**A. SIEM Technology and Tools:** SIEM is a security information event management technology that collects all the infrastructure logs in a centralized manner and helps to organizations to monitor the log activities and increase security. Some popular tools in these technologies mostly used by organizations are Microfocus ArcSight, IBM QRadar, Splunk, and Azure Sentinel.

SIEM tools offer several key features that enhance an organization's security posture:

- Centralized Log Management: SIEM tools consolidate all logs and related information in specific locations making it easier for log management as well as analysis.

- Real-Time Monitoring and Alerts: SIEM tools enable constant analysis of log data for specific patterns and deviation from them and passing the information to the security team in real-time (Ünal et al., 2021).
- Compliance Reporting: The SIEM tools are useful in ensuring that an organization complies with the set regulations by producing compliance reports and keeping records of activities (González-Granadillo, González-Zarzosa and Diaz, 2021).
- Incident Response Support: SIEM tools help in the incident response process by giving information about the security events that has happened to help in the investigation and response processes (Ünal et al., 2021).

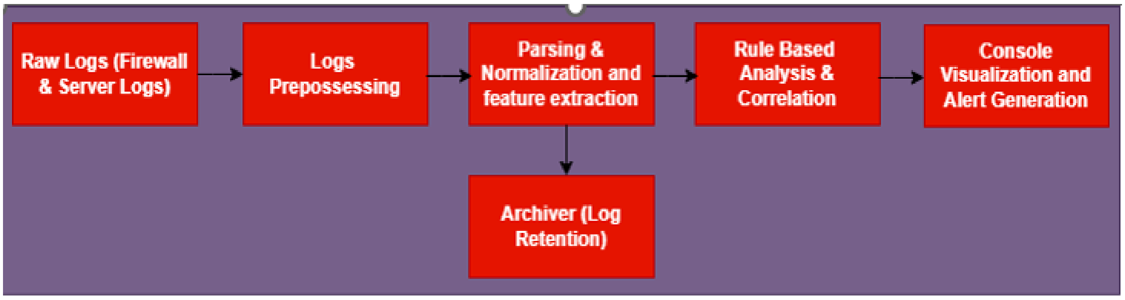

Raw Logs (Firewall & Server Logs)
Logs Prepossessing
Parsing & Normalization and feature extraction
Rule Based Analysis & Correlation
Console Visualization and Alert Generation
Archiver (Log Retention)

Fig: SIEM Log Processing

**B. Machine Learning Approaches in Log Analysis:** Machine learning has affected many fields including cybersecurity by providing new ways of handling large amounts of data and identifying patterns that might be overlooked by other methods. Log analysis is a good example of where machine learning provides powerful tools for anomaly detection, forecasting, and automating the discovery of intricate patterns in log data (Yang et al., 2021).

Machine learning in cybersecurity is when algorithms are created to learn from past information to look for patterns and deviations from the norm. This type of system is more versatile compared to rule-based systems because it does not need new signatures and patterns for new and emerging threats; the machine learning models learn from the new data (Churcher et al., 2021). This flexibility makes machine learning especially useful to improve log analysis.

The structure of the paper helps to address the research objective with the following sections: Section I introduces cyber security with SIEM technology and machine Learning benefits in the log analysis. Section II contains the literature review in this we focus on the problem statement which is the false positive issue in log analysis. Section III contains the Methodology here we are using the isolation forest algorithm to reduce the false positives issue in log analysis. Section IV holds the Results and Discussion including the Isolation Forest algorithm benefits in log analysis and comparing the algorithm benefits with popular SIEM tools and Section V introduces the Conclusion and future research objective.

## II. Literature Review

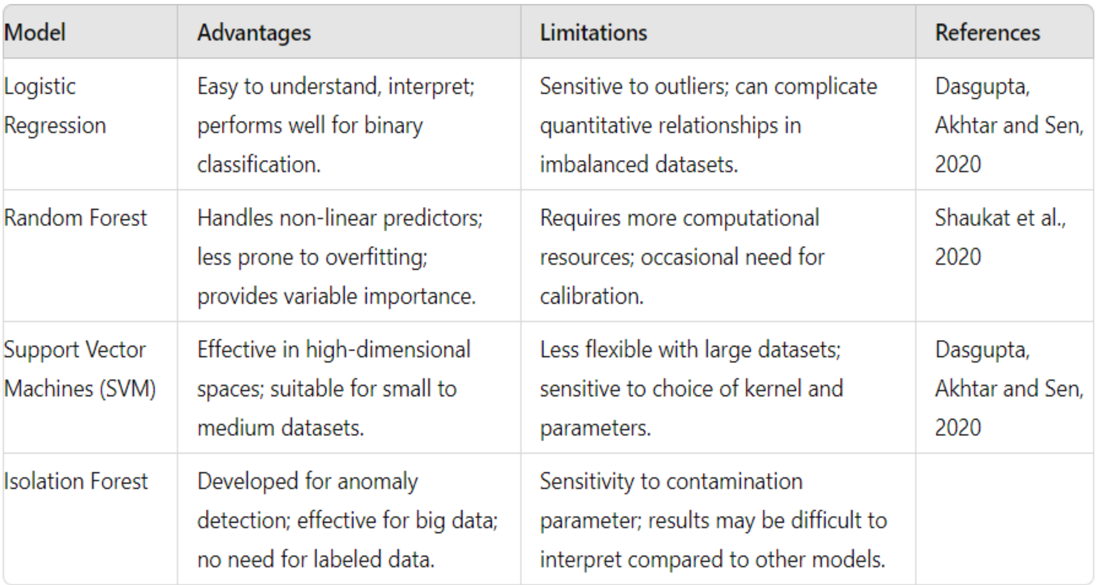
Logs are a crucial component of cybersecurity, and the process referred to as the analysis of logs means the study of the log files generated by the systems, applications, and devices connected to an organization's computing environment. Log files comprise events and activities long, users' login, file and network connections, and system faults (Landauer et al., 2020). These are valuable as the source to get the logs for understanding the organization's functioning and possible breaches of security or non-compliance with certain standards.

At first, it was time-consuming and bulky relying sometimes on a security analyst to perform the analysis on the logs. Nevertheless, as the amount and the variety of logs continuously grows, the process of their analysis becomes extremely time-consuming and ineffective. The present-day log analysis tools like the Security Information and Event Management (SIEM) systems, these a retools that assist in the collection, storage and analysis of log data (Svacina et al., 2020).

It is also important to stress that the logs analysis is to be considered as one of the essential steps of the work. They are applied as the primary means of the security threats identification and enable organizations to timely respond to the given incidents and threats (Yohanandhan et al., 2020).

**A. Problem Statement:** SIEM tools mainly function on the idea of correlation rules as well as signatures and for that reason cannot discover contemporary dynamic and innovative problematic situations. Consequently, organizations are quite often experiencing high false positive rates that lead to alert fatigue therefore, the real threats can easily be overseen. Thus, it has become rather urgent to design more effective and complex approaches to logs analysis due to the enhancement of cyber threats. The earlier tools of system or Security Information and Event Management are more static in their approach which is not sufficient to be dynamic enough and accurate enough to tackle these new threats. Therefore, this research seeks to address these gaps on the integration of machine learning models and the SIEM tools. This integration will leverage the adaptability and data learning capability of the chosen machine learning culture for enhancing getting better of anomaly detection, reducing disappointments of false positive logs, and enhancing the use of log analysis in cybersecurity (Naidu et al., 2022).

**B. Comparison of Various Machine Learning Models for Anomaly Detection:** There are various machine learning models which can be used for log analysis and anomalies detection. Below is the comparison.



| Model | Advantages | Limitations | References |
|---|---|---|---|
| Logistic Regression | Easy to understand, interpret; performs well for binary classification. | Sensitive to outliers; can complicate quantitative relationships in imbalanced datasets. | Dasgupta, Akhtar and Sen, 2020 |
| Random Forest | Handles non-linear predictors; less prone to overfitting; provides variable importance. | Requires more computational resources; occasional need for calibration. | Shaukat et al., 2020 |
| Support Vector Machines (SVM) | Effective in high-dimensional spaces; suitable for small to medium datasets. | Less flexible with large datasets; sensitive to choice of kernel and parameters. | Dasgupta, Akhtar and Sen, 2020 |
| Isolation Forest | Developed for anomaly detection; effective for big data; no need for labeled data. | Sensitivity to contamination parameter; results may be difficult to interpret compared to other models. | |

**Fig: ML Models Comparison Table**

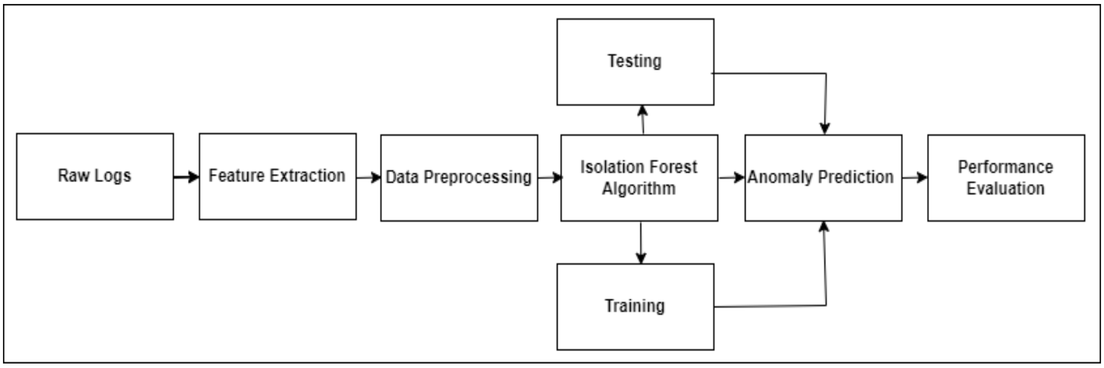


**Fig: Isolation Forest Flow Diagram**

**C. Integration Benefits of Machine Learning with SIEM Tools:** This is especially the case when applying machine learning models together with Security Information and Event Management (SIEM). This integration expands SIEM systems in terms of employing such adaptive, data-driven approaches toward log analysis as to raise the accuracy and efficiency of the process. With machine learning, patterns can be learned and data can be analyzed for anomalies in a way that is near real-time and on a much larger scale as compared to traditional tools for SIEM, thus they are able to decrease the false positives while at the same increasing the detection of more complex threats, as described in Muhammad, Sukarno and Wardana, (2023). This could be missed by the traditional SIEM tools, hence the synergy of the two technologies offer a more holistic, intelligent and scalable approach to threat detection and incident handling hence boosting an organizations security posture.

**D. Objective of the Research:** Evaluate the Limitations of Traditional SIEM Tools: Examine how current traditional systems are ineffective in addressing the volume and variety of log data including high false positives and poor analysts' ability to discover new threats.

Demonstrate the Benefits of SIEM and Machine Learning Integration: Emphasize why it is crucial to combine SIEM tools with machine learning models and how it can enhance the log analysis function and make it faster with a more intelligent approach to threat identification.

Develop a Machine Learning-Based Approach for Anomaly Detection: Of course, it is necessary to train a machine learning model for this specific purpose of detecting anomalies only in the fields of firewall log data and demonstrating and proving its usage and effectiveness.

Compare Performance with Traditional Methods: In the comparison, highlight the efficiency of the proposed machine learning-based approach against the traditional SIEM tools based on the metrics, including accuracy, false positive rates, and the capacity to recognize complex threats.

**E. Significance of the Study**: The relevance of this study to cybersecurity as a field is the following. Firstly, it contributes to the existing research as it presents results on the impact of machine learning integration to SIEM tools. Thus, while presenting positive outcomes that result from such integration, the research provides a beneficial strategic approach toward improving threat identification.

Secondly, the study has relevance in practice for organizations that aim at strengthening their protection against cyber threats. As the sophistication of threats rises, conventional SIEM solutions do not suffice. This incorporation of machine learning models into the traditional signature-based recognition can result to an inclusion of smart and adaptive analysis of the data, therefore decreasing on the number of false alarms and enhancing recognition of on new threats (Aljabri et al., 2022).

On a final note, the research also points towards a paradigm shift of a great extent on cybersecurity measures. Thus, in terms of goals set for the study, this study shows how the existing problem with the lack of an effective and adequate threat detection system can be solved by the integration of machine learning into the existing SIEM framework and help to create a stronger security system for any organization.

## III. Proposed Methodology

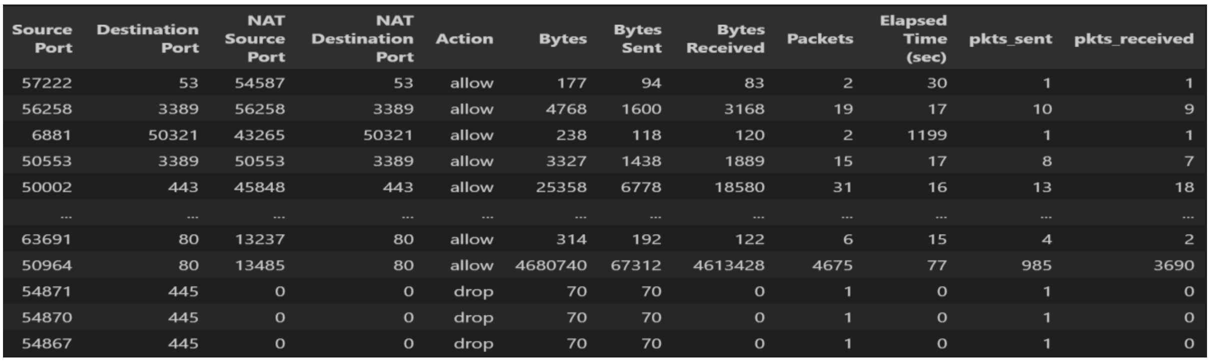
In this section, The data was collected from a mid-size enterprise's firewall logs which are responsible for recording all the activities in a network ranging from connections, data transfer and security incidents and including various network activities for several months. Later a Machine learning algorithm was performed on the data for anomaly detection. The data set is very important for the analysis and identification of abnormal situations that may signal a threat to security (Chen et al., 2021).

Some of the features that are encompassed in this vast database are as follows: They help in analyzing the behavior of the network and in detecting any irregularities.

**A. Structure and Characteristics of the Data:** The dataset consists of multiple features, each representing a specific aspect of network activity:

- **Source Port:** The port number on the source machine initiating the connection.
- **Destination Port:** The port number on the destination machine receiving the connection.
- **NAT Source Port:** The source port number after Network Address Translation (NAT).
- **NAT Destination Port:** The destination port number after NAT.
- **Action:** Indicates whether the connection was allowed or denied by the firewall.
- **Bytes:** Total bytes transferred during the connection.

- **Bytes Sent:** Bytes sent from the source to the destination.
- **Bytes Received:** Bytes received by the source from the destination.
- **Packets:** Total number of packets transmitted during the connection.
- **Elapsed Time (sec):** Duration of the connection in seconds.
- **Packets Sent:** Number of packets sent by the source.
- **Packets Received:** Number of packets received by the source.



| Source Port | Destination Port | NAT Source Port | NAT Destination Port | Action | Bytes | Bytes Sent | Bytes Received | Packets | Elapsed Time (sec) | pkts_sent | pkts_received |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 57222 | 53 | 54587 | 53 | allow | 177 | 94 | 83 | 2 | 30 | 1 | 1 |
| 56258 | 3389 | 56258 | 3389 | allow | 4768 | 1600 | 3168 | 19 | 17 | 10 | 9 |
| 6881 | 50321 | 43265 | 50321 | allow | 238 | 118 | 120 | 2 | 1199 | 1 | 1 |
| 50553 | 3389 | 50553 | 3389 | allow | 3327 | 1438 | 1889 | 15 | 17 | 8 | 7 |
| 50002 | 443 | 45848 | 443 | allow | 25358 | 6778 | 18580 | 31 | 16 | 13 | 18 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 63691 | 80 | 13237 | 80 | allow | 314 | 192 | 122 | 6 | 15 | 4 | 2 |
| 50964 | 80 | 13485 | 80 | allow | 4680740 | 67312 | 4613428 | 4675 | 77 | 985 | 3690 |
| 54871 | 445 | 0 | 0 | drop | 70 | 70 | 0 | 1 | 0 | 1 | 0 |
| 54870 | 445 | 0 | 0 | drop | 70 | 70 | 0 | 1 | 0 | 1 | 0 |
| 54867 | 445 | 0 | 0 | drop | 70 | 70 | 0 | 1 | 0 | 1 | 0 |

**Figure 1: Dataset sample.**

**B. Data Collection Process:** The log data was collected from the organization's firewall system as it provides real-time monitoring of all the traffic on the network. They were also pre-processed and anonymized to exclude any sensitive information from being used in the analysis.

**C. Data Preprocessing and Feature Engineering:** The first of the data preprocessing techniques involved data cleaning where missing values were located and dealt with. Issues of missing data are critical in machine learning because missing data can alter the accuracy of the model. Concerning the issue of missing values in the dataset, any record with a missing value was scrutinized (Li et al., 2020). The type of data collected from the organization's firewall logs was nearly comprehensive, as few missing values were discovered.

**D. Creation of New Features:** Feature transformation on the other hand involves developing new features from the present features in order to extract more information that cannot be obtained from the raw data. In this study, several new features were created:

- **Bytes per Packet:** Defined as the ratio of the total bytes to the total number of packets. This particular feature indicates the average size of packets transmitted during the test.
- **Elapsed Time per Packet:** Expressed as the elapsed time divided by the total number of packets, it provides the average time per packet.
- **Sent to Received Bytes Ratio:** The send bytes over receive bytes ratio which can be used to make some comparisons and identify some oddities.

**E. Model Selection and Training:** The proper model selection and training of the machine learning model contributes to better anomaly detection with considerable accuracy. In this study, the Isolation Forest algorithm was selected based on the results of the previous studies and its ability to work with high-dimensional data and being specifically developed for anomaly detection (Guo, Yuan and Wu, 2021).

**F. Criteria for Model Selection:** The Isolation Forest algorithm was selected based on several key criteria:

**Scalability:** Its capacity to deal with a sizeable amount of log data which is particularly suitable since the organization has a massive amount of data from the firewall logs.

**Accuracy:** Its capacity to demonstrate high efficiency when it comes to pre-screening the anomalies using the randomly generated decision trees thus minimizing on the occurrence of false positives and false negatives (Tiwari, 2019).

**Computational Efficiency:** The Isolation Forest is faster, and as a result, is well suitable for online log analysis and security threats detection (Tiwari, 2019).

**G. Training Process and Parameters:** The training process can be a complex task depending on the required effectiveness of the model. First, the dataset was divided into the training and testing set so as to get a good conclusion on the fitness of the model. Hyperparameter tuning was also done via cross-validation to avoid creating a model that learns the noise in the data so this will be ready to generalize with unseen data sets (Herath, 2024).

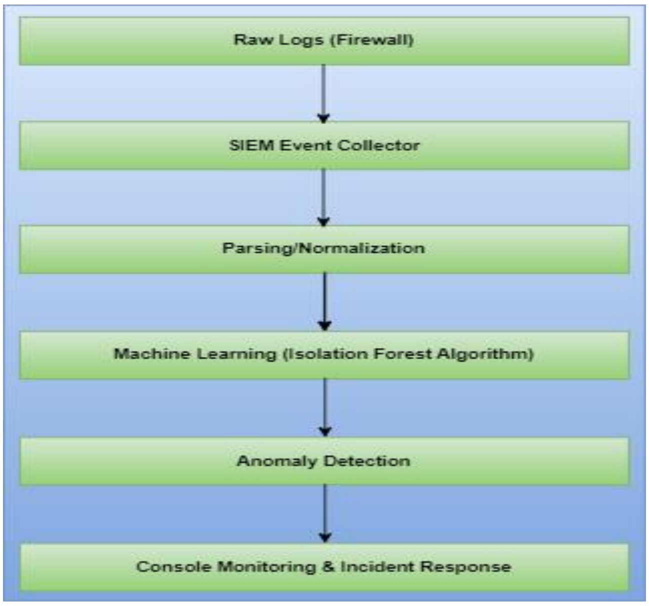
**H. Evaluation Metrics:** In order to evaluate the outcome of the Isolation Forest model on the firewall log data for the purpose of identifying anomalies, evaluation techniques were used.

Accuracy focuses on the general correctness of the model, which is the ratio of the number of true positives and true negatives to all the cases predicted. While accuracy is informative, it is not suitable to be used in cases of anomaly detection because of the skewed distribution of data where the anomalous cases are a lot fewer than the normal cases (Nath, 2023).

Precision is concerned with the number of correct positive instances identified from the total number of instances flagged as positive by the model. Sensitivity or recall on the other hand, calculates the ratio of true positive cases to all the actual anomalous cases. High recall also means that the model is able to detect most of the true anomalies and does not miss many of them. F1 Score is the average of the precision and recall and is considered to be a better measure as it has a balance between the two. It is more useful when the data set is skewed, and it takes into consideration both the false positive and the false negative cases (Retinraj, 2023).

**I. Proposed Architecture:** Proposed Method Steps:

- Network traffic passes through the firewall.
- The firewall integrates with SIEM Event Collector for logging.
- SIEM parses and normalizes the collected logs.
- The Isolation Forest Algorithm analyzes the traffic data.
- The algorithm distinguishes normal instances and anomalies.
- Alerts are generated accordingly.


Raw Logs (Firewall)
SIEM Event Collector
Parsing/Normalization
Machine Learning (Isolation Forest Algorithm)
Anomaly Detection
Console Monitoring & Incident Response

**J. Implementation Details:** In the case of anomaly detection by the machine learning-based approach, several steps were followed, and the tools and libraries were used to support the process. The main tools applied were Python and its libraries, namely Pandas for data processing, Scikit-learn for machine learning algorithms, and Matplotlib and Seaborn for data visualization (Saxena, 2020).

The first step for the dataset was to load it and perform preprocessing using the Pandas library which also helped in the handling of missing values, normalizing the data and feature engineering. As for the Anomaly Detection method, the Isolation Forest model which is appropriate for this kind of task was chosen and applied with the help of Scikit-learn. Hyperparameter tuning was done using GridSearchCV to find out the best parameters to use for the number of estimators and contamination rate (Saxena, 2020).

## IV. Results & Discussion

In this section, we discussed the outcomes of this methodology. This algorithm finds the anomalies in a better way as compared to the traditional SIEM tools.

**A. Exploratory Data Analysis (EDA):** Advanced data analysis begins with the Exploratory Data Analysis (EDA) to know the characteristics of the data set and to look for hidden patterns. Regarding the initial data analysis, EDA was conducted on the firewall log data to understand the distribution of the data as well as the correlation between the features.

### B. Data Distribution and Descriptive Statistics:

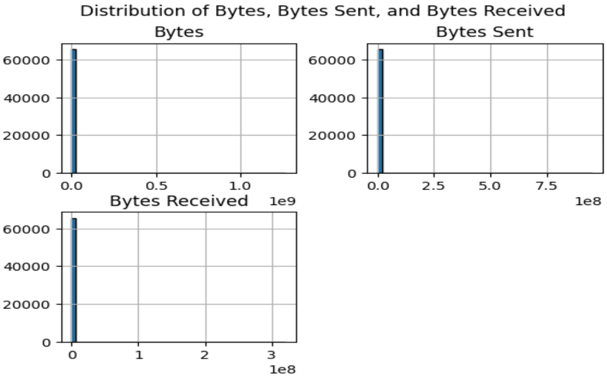


*Figure 2: Distribution of data.*

The first procedure included the study of the distribution of essential characteristics like Bytes, Bytes Sent, and Bytes Received. These features are depicted in histograms and they show that they have heavy-tailed distributions with most of the values close to zero. This skewness is normal in logs in which a few connections may represent large data traffic while most represent small traffic.

### C. Visualization of Key Features:

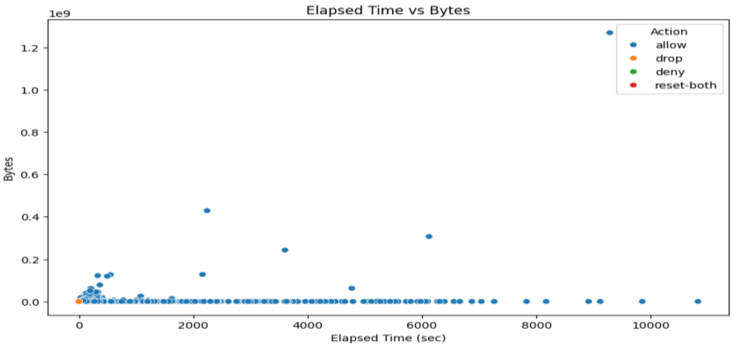
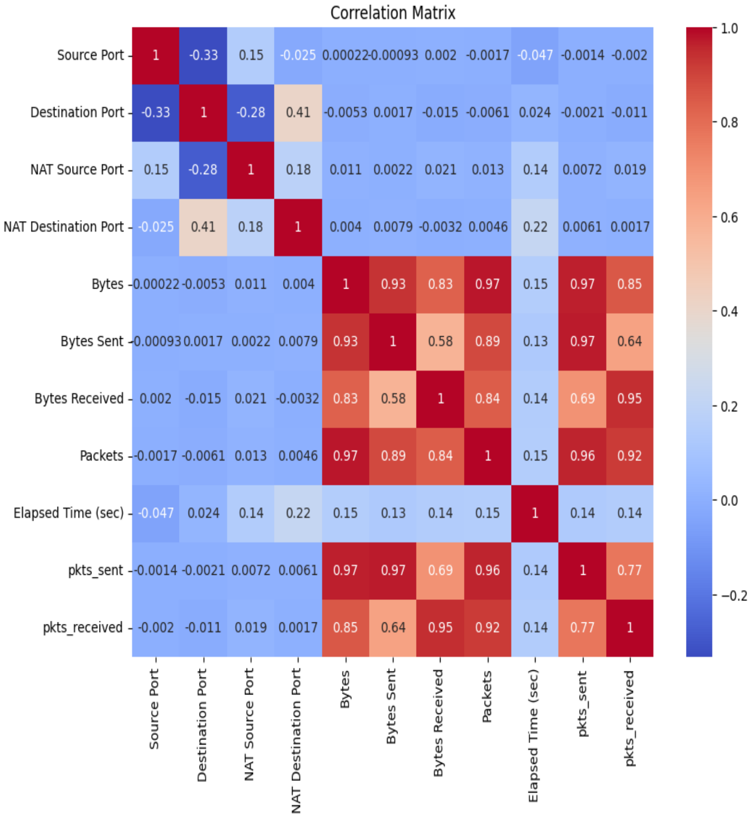


**Figure 3: Scatter plot for elapsed time vs bytes.**

The second graph is the scatter plot of Elapsed Time (sec) and Bytes where more information can be gathered. The plot shows that most of the points are located around the origin, which means short durations and low byte transfers, but some points are considerably different. These may be outliers because of certain activities that are quite different from the other observations and thus may be worth investigating further (Quatrini et al., 2020).



| | Source Port | Destination Port | NAT Source Port | NAT Destination Port | Bytes | Bytes Sent | Bytes Received | Packets | Elapsed Time (sec) | pkts_sent | pkts_received |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Port | 1 | -0.33 | 0.15 | -0.025 | 0.00022 | -0.00093 | 0.002 | -0.0017 | -0.047 | -0.0014 | -0.002 |
| Destination Port | -0.33 | 1 | -0.28 | 0.41 | -0.0053 | 0.0017 | -0.015 | -0.0061 | 0.024 | -0.0021 | -0.011 |
| NAT Source Port | 0.15 | -0.28 | 1 | 0.18 | 0.011 | 0.0022 | 0.021 | 0.013 | 0.14 | 0.0072 | 0.019 |
| NAT Destination Port | -0.025 | 0.41 | 0.18 | 1 | 0.004 | 0.0079 | -0.0032 | 0.0046 | 0.22 | 0.0061 | 0.0017 |
| Bytes | 0.00022 | -0.0053 | 0.011 | 0.004 | 1 | 0.93 | 0.83 | 0.97 | 0.15 | 0.97 | 0.85 |
| Bytes Sent | -0.00093 | 0.0017 | 0.0022 | 0.0079 | 0.93 | 1 | 0.58 | 0.89 | 0.13 | 0.97 | 0.64 |
| Bytes Received | 0.002 | -0.015 | 0.021 | -0.0032 | 0.83 | 0.58 | 1 | 0.84 | 0.14 | 0.69 | 0.95 |
| Packets | -0.0017 | -0.0061 | 0.013 | 0.0046 | 0.97 | 0.89 | 0.84 | 1 | 0.15 | 0.96 | 0.92 |
| Elapsed Time (sec) | -0.047 | 0.024 | 0.14 | 0.22 | 0.15 | 0.13 | 0.14 | 0.15 | 1 | 0.14 | 0.14 |
| pkts_sent | -0.0014 | -0.0021 | 0.0072 | 0.0061 | 0.97 | 0.97 | 0.69 | 0.96 | 0.14 | 1 | 0.77 |
| pkts_received | -0.002 | -0.011 | 0.019 | 0.0017 | 0.85 | 0.64 | 0.95 | 0.92 | 0.14 | 0.77 | 1 |

**Figure 4: Correlation matrix.**

The heatmap of the correlation matrix shows the interaction between different features of the dataset. Specifically, Bytes, Bytes Sent, Bytes Received, and Packets are positively correlated, which means that connections with larger byte transfers are likely to have more packets. However, features like Source Port and Destination Port are less correlated with the other variables which suggests they are less useful in anomaly detection (Wang et al., 2020).

**D. Initial Observations and Insights:** From the EDA, several key insights emerged:

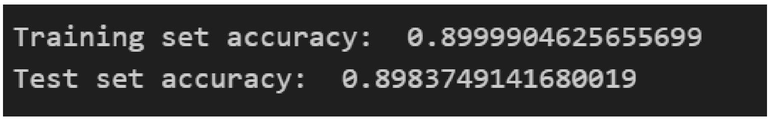
**Skewed Distributions:** From the descriptions of byte-related features, it can be seen that the distributions are highly skewed, and therefore, normalization and standardization of the features during data preprocessing are crucial.

**Outliers:** Scatter plots that were created display outliers that point to potential anomalies and this is in line with the objectives of the study to improve on the detection of anomalies.

**Feature Correlations:** The high degree of dependency between some of the features can be useful in feature selection and feature engineering to use the right variables in the model.

In summary, the EDA gave a general overview of the firewall log data, which helped direct the preprocessing, feature engineering, and training of the models. These insights are important to building a better machine-learning solution for enhancing log analysis and anomaly detection in SIEM systems.

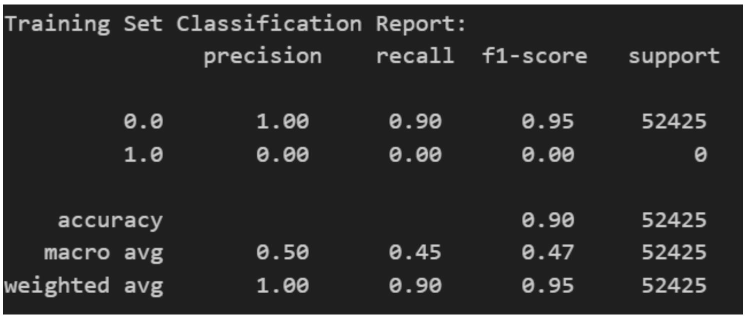
**E. Model training and testing results:** The features of the Isolation Forest model were tested using the firewall log dataset, with the division into training and testing data. Other performance indicators such as accuracy, precision, recall, and F1-score were used to evaluate the efficiency of the model.



```
Training set accuracy:  0.8999904625655699
Test set accuracy:  0.8983749141680019
```

Figure 5: Training and testing accuracy.

**F. Performance of the model on training data:**

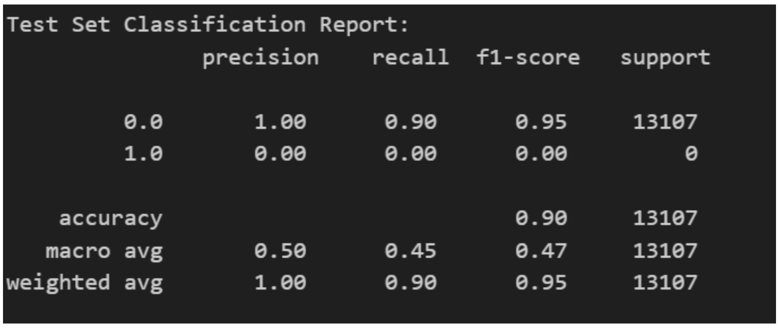


```
Training Set Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.90      0.95     52425
         1.0       0.00      0.00      0.00         0

    accuracy                           0.90     52425
   macro avg       0.50      0.45      0.47     52425
weighted avg       1.00      0.90      0.95     52425
```

**Figure 6: Training set classification report.**

The Isolation Forest model was built on the data set which was pre-processed and for which features were engineered. The accuracy obtained on the training set was roughly about 89 percent. The normal instances that were obtained from the normal distribution of the data were also classified with 99% accuracy by the model. However, the training set confusion matrix and classification report raise a very important observation (Elmrabit et al., 2020). From the confusion matrix, the model accurately classified 47182 normal instances as normal class and misclassified 5243 normal instances as anomalies. This was evident from the classification report that showed the precision was 1.00, recall of 0.90, and F1-score of 0.95 for normal instances.

**G. Model Performance on Test Data:**



```
Test Set Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.90      0.95     13107
         1.0       0.00      0.00      0.00         0

    accuracy                           0.90     13107
   macro avg       0.50      0.45      0.47     13107
weighted avg       1.00      0.90      0.95     13107
```

**Figure 7: Testing dataset classification report.**

The performance of the model was then tested on the test set achieving an accuracy of about 89 percent. 84%. Likewise with the training set, the test set outcomes present difficulties in anomaly identification. From the confusion matrix of the test set, it was seen that 11775 normal instances were correctly classified as normal (TN) and 1332 normal instances were wrongly classified as anomalies (FP). The model was again able to correctly predict no anomalies, meaning no true positives and false negatives. The precision of the test set according to the classification report was 1.00, recall of 0.90 (Al-amri et al., 2021). These results corroborate the findings during training; while the model predicts normal cases with reasonable accuracy, it struggles to classify anomalies.

### H. Anomaly Detection Analysis:

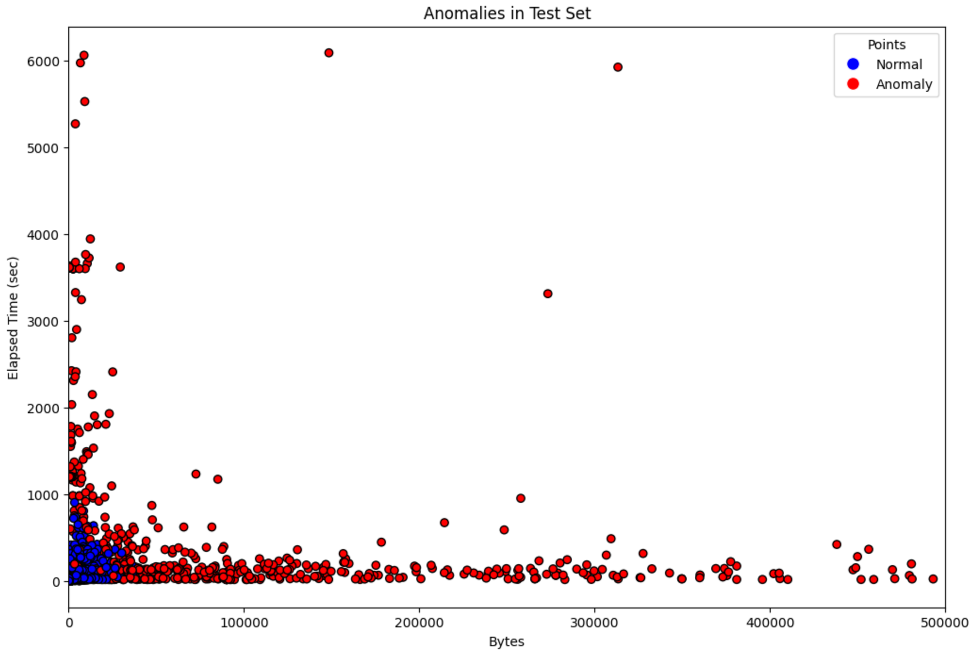


Figure 8: Anomalies and normal logs detected.

The results of the Isolation Forest model in detecting anomalies were further examined through the use of figures and measures of central tendency. The scatter plot of the anomalies in the test set presented in Figure 10 is characterized by a large number of red dots, indicating anomalies, distributed across the range of byte transfers and the elapsed time. The majority of data points are located in the region surrounding the origin, meaning that short-lasting and low-byte exchange is the norm. However, the model highlighted some points that have higher byte transfer and longer time as the outliers (Nedelkoski et al., 2020).

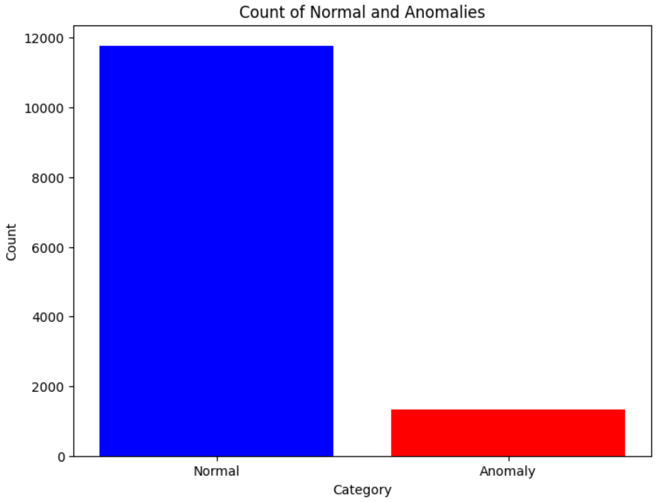


Figure 9: Bar chart showing normal vs anomaly log data.

The bar chart seen in figure 11 below shows the distribution of normal instances as well as the anomalies in the test set. As can be observed from the chart, there are many more normal cases (blue) compared to the anomalies (red). This situation highlights the issues related to the problem of an unequal distribution of instances in anomaly detection problems.

**I. Interpretation of Results:** The analysis and conclusions of this research will help to determine the strengths and weaknesses of the application of the Isolation Forest model for the analysis of anomalies in the firewall log data. These findings were gleaned on the basis of the following observations of the key results:

The model has a high overall accuracy with regard to identifying normal instances; it, however, has a low accuracy when it comes to the identification of anomalies (Shaukat et al., 2020).

The training set accuracy of about 89 percent is implied because it is the fraction that was correctly classified out of the entire set. test set accuracy of 89. 84% of them are typical network traffic and it shows the high effectiveness of the chosen model. However, the comparative analysis of the confusion matrices and classification reports shows severe shortcomings. The precision, recall as well as the F1-score for anomalies were all at zero implying that the model did not manage to detect any of the anomalies.

Comparing these results with the literature, there are similar works that have indicated the issues of detecting anomalies in imbalanced datasets. Rule-based conventional SIEM solutions also have problems with identification of new and complex threats and have a large number of false positives (Yadav, Kumar and Dhavale, 2020). To overcome these issues, the study incorporates machine learning, especially the unsupervised learning model Isolation Forest which does not require the sample anomalies. That being the case, the outcomes of this study correspond to the more extensive problem of training models on extremely skewed distributions.

**J. Implications of the Findings:** The general impact of the results of this study on the field of cybersecurity is quite profound, and in particular, in the continued development of the existing SIEM tools through the application of machine learning techniques. The results of the Isolation Forest model are relatively accurate and precise, which can affirm the decision-makers to apply this model as an additional component to further develop the typical SIEM systems. This integration is a more innovative form of analyzing logs as it can be modified in line with the rising dangers in the realm of computer crime (Choi et al., 2021).

Due to the high accuracy of the Isolation Forest model, the network activities can be well detected and the chances of the model failing to detect potential threats are very slim. Thus, the ability to distinguish between ordinary and potentially malicious transactions serves as a reliable method of threat recognition that is necessary to maintain a safe space on the Internet. This capability is much better than the rule-based systems that were used earlier because they rely on the previous patterns that a newcomer or an innovator can easily overlook (Stoian, 2020).

**K. Benefits of Isolation Forest Algorithm:** As a result of the Isolation Forest model analysis, it is possible to conclude that the positive prospects for the further development and enhancement of SIEM tools are to achieve high accuracy in the identification of anomalies, as well as the possibility of processing large amounts of data. Compared to competitors like Splunk, IBM QRadar, and ArcSight, the Isolation Forest model has fewer false positives and integration possibilities (Muhammad, Sukarno and Wardana, 2023). Traditional SIEM tools also have good Real-time analysis and Advanced analytics but these tools require a large amount of hardware and software resources and hence expensive. Implementing machine learning algorithms such as the Isolation Forest can be a much more cost-effective and a scalable solution, which will drastically enhance the effectiveness and practicability of the SIEM systems (Kilincer, Ertam and Sengur, 2021).

**L. Comparison with SIEM tools:**

| Aspect | Isolation Forest Model | Splunk | IBM QRadar | ArcSight |
|---|---|---|---|---|
| **Anomaly Detection** | High accuracy in identifying anomalies | Effective with advanced analytics | Robust detection with AI integration | Strong anomaly detection capabilities |
| **Scalability** | Highly scalable, handles large datasets well | Scalable but resource-intensive | Highly scalable with distributed design | Scalable, suitable for large enterprises |

| | | | | |
|---|---|---|---|---|
| **Ease of Integration** | Seamless integration with existing SIEM tools | Complex setup, requires expertise | Integrated solution with AI features | Comprehensive integration but complex |
| **False Positive Rate** | Low false positive rate | Moderate, dependent on rule configurations | Low, with adaptive learning features | Moderate, with customizable rule sets |
| **Real-time Analysis** | Capable of real-time anomaly detection | Strong real-time capabilities | Real-time monitoring and analytics | Real-time event correlation |
| **Cost Efficiency** | Cost-effective, uses open-source libraries | Higher cost due to licensing fees | Higher cost with advanced features | High initial cost and maintenance fees |

**M. Summary of Key Findings:** As for this research, the objective was to determine the possibilities of combining machine learning models, such as the Isolation Forest algorithm, with conventional SIEM tools to improve the analysis of logs and the identification of outliers. Some of the goals were to understand the drawbacks of conventional SIEM solutions, show the advantages of integrating the proposed solution, establish a machine learning approach for anomaly detection, and compare its effectiveness with the others (Wang et al., 2020). This study suggests that the proposed solution of applying the Isolation Forest model to the SIEM system can lead to an increase in the accuracy of threat detection and the efficiency of such a system. The model has attained satisfactory generalization with the accuracies of the training and the test set being approximately 89.99% and 89.84%, respectively. The above results clearly indicate that the proposed model is capable of processing a large amount of log data and also can differentiate between normal and malicious network traffic. These key insights corroborate that combining machine learning with SIEM tools can help enhance the analytical capabilities of these tools and thus increase threat detection.

**N. Contributions of the Research**: The following are some of the research contributions that can be highlighted to the body of knowledge in cybersecurity: Firstly, it provides insights about how the machine learning models can be introduced into the functions of SIEM tools to enhance the log review procedure. The study proves that not only can the Isolation Forest model integrate with big data but also the offered model is highly effective in identifying patterns and exceptions most of the time neglected by rule-based techniques The given research also supports the idea that the development of new methods in log analysis, such as Isolation Forest model, promotes a step forward in the progression of old approaches (Shaukat et al., 2020).

Secondly, the research is conducted to analyse the precise potential of adopting machine learning for the enhancement of SIEM tools. Isolation Forest used in the SIEM system as the model enables to enhance the work's efficiency, reduce the number of false positive outcomes, and enhance threat recognition. It has implications to real-word specifically to the organizations and their security teams, as this improvement helps them to filter what is real and concentrate on it as well as prepare for different security situations.

## V. Conclusion & Future Work

The isolation Forest algorithm reduced the false positive counts but In the normal status of the model, the recognition rate is very high, but for the anomalies, there are very few true positives. This is a big disadvantage, especially in real-life situations for which the ability to tag out anomalies in the correct manner is paramount in a bid to prevent and contain threats in the early stages (Poornima and Paramasivan, 2020). These include the false positive and negative rates and the use of the visualization table to get some insight into the model and its drawbacks. Future work may define methods of approaching the data imbalance problem and study the other approaches to make the resulting model more sensitive to anomalous items.

## References

[1] Cao, Q., Qiao, Y. and Lyu, Z. (2017). Machine learning to detect anomalies in web log analysis. [online] IEEE Xplore. doi:https://doi.org/10.1109/CompComm.2017.8322600.

[2] He, S., He, P., Chen, Z., Yang, T., Su, Y. and Lyu, M.R. (2021). A Survey on Automated Log Analysis for Reliability Engineering. ACM Computing Surveys, 54(6), pp.1–37. doi:https://doi.org/10.1145/3460345.

[3] Yadav, R.B., Kumar, P.S. and Dhavale, S.V. (2020). A Survey on Log Anomaly Detection using Deep Learning. 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). doi:https://doi.org/10.1109/icrito48877.2020.9197818.

[4] Ünal, U., Kahya, C.N., Kurtlutepe, Y. and Dağ, H. (2021). Investigation of Cyber Situation Awareness via SIEM tools: a constructive review. [online] IEEE Xplore. doi:https://doi.org/10.1109/UBMK52708.2021.9558964.

[5] González-Granadillo, G., González-Zarzosa, S. and Diaz, R. (2021). Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. Sensors, [online] 21(14), p.4759. doi:https://doi.org/10.3390/s21144759.

[6] Yang, L., Chen, J., Wang, Z., Wang, W., Jiang, J., Dong, X. and Zhang, W. (2021). Semi-Supervised Log-Based Anomaly Detection via Probabilistic Label Estimation. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICSE43902.2021.00130.

[7] Churcher, A., Ullah, R., Ahmad, J., ur Rehman, S., Masood, F., Gogate, M., Alqahtani, F., Nour, B. and Buchanan, W.J. (2021). An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks. Sensors, [online] 21(2), p.446. doi:https://doi.org/10.3390/s21020446.

[8] Landauer, M., Skopik, F., Wurzenberger, M. and Rauber, A. (2020). System log clustering approaches for cyber security applications: A survey. Computers & Security, 92, p.101739. doi:https://doi.org/10.1016/j.cose.2020.101739.

[9] Svacina, J., Raffety, J., Woodahl, C., Stone, B., Cerny, T., Bures, M., Shin, D., Frajtak, K. and Tisnovsky, P. (2020). On Vulnerability and Security Log analysis. Proceedings of the International Conference on Research in Adaptive and Convergent Systems. doi:https://doi.org/10.1145/3400286.3418261.

[10] Yohanandhan, R.V., Elavarasan, R.M., Manoharan, P. and Mihet-Popa, L. (2020). Cyber-Physical Power System (CPPS): A Review on Modeling, Simulation, and Analysis With Cyber Security Applications. IEEE Access, 8, pp.151019–151064. doi:https://doi.org/10.1109/access.2020.3016826.

[11] Naidu, K.B., Ravi Prasad, B., Hassen, S.M., Kaur, C., Al Ansari, M.S., Vinod, R., Nivetha, M. and Kiran Bala, B. (2022). Analysis of Hadoop log file in an environment for dynamic detection of threats using machine learning. Measurement: Sensors, 24, p.100545. doi:https://doi.org/10.1016/j.measen.2022.100545.

[12] Muhammad, A.R., Sukarno, P. and Wardana, A.A. (2023). Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning. Procedia Computer Science, [online] 217, pp.1406–1415. doi:https://doi.org/10.1016/j.procs.2022.12.339.

[13] Aljabri, M., Alahmadi, A.A., Mohammad, R.M.A., Aboulnour, M., Alomari, D.M. and Almotiri, S.H. (2022). Classification of Firewall Log Data Using Multiclass Machine Learning Models. Electronics, 11(12), p.1851. doi:https://doi.org/10.3390/electronics11121851.

[14] Chen, Z., Liu, J., Gu, W., Su, Y. and Lyu, M.R. (2021). Experience Report: Deep Learning-based System Log Analysis for Anomaly Detection. [online] arXiv.org. Available at: https://arxiv.org/abs/2107.05908.

[15] Li, X., Chen, P., Jing, L., He, Z. and Yu, G. (2020). SwissLog: Robust and Unified Deep Learning Based Log Anomaly Detection for Diverse Faults. [online] IEEE Xplore. doi:https://doi.org/10.1109/ISSRE5003.2020.00018.

[16] Guo, H., Yuan, S. and Wu, X. (2021). LogBERT: Log Anomaly Detection via BERT. 2021 International Joint Conference on Neural Networks (IJCNN). doi:https://doi.org/10.1109/ijcnn52387.2021.9534113.

[17] Tiwari, S. (2019). Complete Guide to Machine Learning Evaluation Metrics. [online] Medium. Available at: https://medium.com/analytics-vidhya/complete-guide-to-machine-learning-evaluation-metrics-615c2864d916.

[18] Herath, S. (2024). Theoretical Basis of ML — Model Evaluation Metrics(Summary). [online] Data Science and Machine Learning. Available at: https://medium.com/image-processing-with-python/theoretical-basis-of-ml-model-evaluation-metrics-summary-3cae19129679.

[19] Nath, S. (2023). Model Evaluation Metrics: A Comprehensive Guide. [online] Medium. Available at: https://medium.com/@sruthy.sn91/model-evaluation-metrics-a-comprehensive-guide-96b71c732937.

[20] Retinraj (2023). Machine Learning System Design Stage: Model Evaluation. [online] Medium. Available at: https://pauldeepakraj-r.medium.com/machine-learning-system-design-stage-model-evaluation-4b7c78f1ea0b [Accessed 10 Jul. 2024].

[21] Saxena, A. (2020). Python Libraries. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/python-libraries-d73859384c43 [Accessed 10 Jul. 2024].

[22] Quatrini, E., Costantino, F., Di Gravio, G. and Patriarca, R. (2020). Machine learning for anomaly detection and process phase classification to improve safety and maintenance activities. Journal of Manufacturing Systems, 56, pp.117–132. doi:https://doi.org/10.1016/j.jmsy.2020.05.013.

[23] Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P.K., Alfarraj, O. and Tolba, A. (2020). LogEvent2vec: LogEvent-to-Vector Based Anomaly Detection for Large-Scale Logs in Internet of Things. Sensors, 20(9), p.2451. doi:https://doi.org/10.3390/s20092451.

[24] Elmrabit, N., Zhou, F., Li, F. and Zhou, H. (2020). Evaluation of Machine Learning Algorithms for Anomaly Detection. [online] IEEE Xplore. doi:https://doi.org/10.1109/CyberSecurity49315.2020.9138871.

[25] Al-amri, R., Murugesan, R.K., Man, M., Abdulateef, A.F., Al-Sharafi, M.A. and Alkahtani, A.A. (2021). A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data. Applied Sciences, [online] 11(12), p.5320. doi:https://doi.org/10.3390/app11125320.

[26] Nedelkoski, S., Bogatinovski, J., Acker, A., Cardoso, J. and Kao, O. (2020). Self-Attentive Classification-Based Anomaly Detection in Unstructured Logs. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICDM50108.2020.00148.

[27] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I.A. and Xu, M. (2020). A Survey on Machine Learning Techniques for Cyber Security in the Last Decade. IEEE Access, 8, pp.222310–222354. doi:https://doi.org/10.1109/access.2020.3041951.

[28] Choi, K., Yi, J., Park, C. and Yoon, S. (2021). Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. IEEE Access, 9, pp.120043–120065. doi:https://doi.org/10.1109/access.2021.3107975.

[29] Stoian, N.A. (2020). Machine Learning for anomaly detection in IoT networks : Malware analysis on the IoT-23 data set. [online] essay.utwente.nl. Available at: https://essay.utwente.nl/81979/.

[30] Kilincer, I.F., Ertam, F. and Sengur, A. (2021). Machine Learning Methods for Cyber Security Intrusion Detection: Datasets and Comparative Study. Computer Networks, [online] 188, p.107840. doi:https://doi.org/10.1016/j.comnet.2021.107840.

[31] Poornima, I.G.A. and Paramasivan, B. (2020). Anomaly detection in wireless sensor network using machine learning algorithm. Computer Communications, 151, pp.331–337. doi:https://doi.org/10.1016/j.comcom.2020.01.005.