[1]Salam Allawi Hussein

[2]Sándor R. Répás

Anomaly Detection in Log Files Based on Machine Learning Techniques

Abstract: - This article provides a comprehensive overview of contemporary techniques for detecting anomalies in log files in light of the growing reliance on computer systems and the volume of log files generated. Log files are crucial for identifying questionable or malicious activities since they shed light on system behavior and performance. The work addresses the challenges associated with identifying anomalies in log files, including their dynamic structure, high volume, and chaotic nature. Several anomaly detection strategies are assessed based on how well they work, how quickly they can be executed, and how well they can be applied to different types of log files. These strategies include statistical techniques, machine learning algorithms, and deep learning techniques. Furthermore, because cyber threats are getting more complex, AI applications are becoming crucial to network and cyber security. By utilizing anomaly detection, predictive analysis, and reactions to adjust to changing attack patterns, artificial intelligence can significantly enhance security.

Keywords: Machine learning, cybersecurity, anomaly detection, log files

I. INTRODUCTION

Anomaly detection (AD), sometimes referred to as outlier or novelty discovery, is a method used to find unusual occurrences or data points in a collection. These abnormalities could yield important information in several fields, such as cybersecurity, healthcare, and telecommunications [1], [2], and [3]. Anomalies, for example, can help discover strange calling or message patterns to signal possible security problems or fraudulent operations, or they can help detect irregularities in medical imaging or vital signs to diagnose illnesses.

The process of data analysis, also referred to as "anomaly detection," aims to find outliers or unusual information in a sample of data. The data mining community is very interested in this branch of research since it entails finding unusual and fascinating patterns in data. It has been thoroughly investigated in machine learning and statistics [4], [5] under several names, including exception mining, deviation analysis, outlier detection, and novelty discovery. Notably, different academic fields have used distinct definitions to characterize anomalies. However, Hawkins's identification of outliers [6] description of anomalies has gained widespread acceptance in the research community." An anomaly is an observation that differs so drastically from others that it raises questions about the mechanism that produced it".

Anomaly detection can be supervised or unsupervised. Supervised anomaly detection uses a model trained on labeled data to discover and categorize anomalies. On the other hand, unsupervised anomaly detection techniques aim to identify patterns in the data that deviate from the norm or are considered unusual without using labeled data [7]. Sure, the latter is preferred since labeled data is relatively easy to acquire.

The choice of the most suitable anomaly detection method depends on the specific properties of the dataset and the type of abnormalities being investigated. Supervised approaches are useful when working with labeled data but can face challenges in cases where obtaining labeled data is time-consuming and unpractical. Unsupervised approaches are useful in situations when abnormalities need to be identified without using labeled data. However, a difficulty appears when deciding what is considered normal based entirely on the provided unlabeled information. [2][5][7].

Modern research on anomaly detection emphasizes the importance of creating reliable approaches to identify various types of anomalies. This provides the framework to develop robust and flexible anomaly detection systems

[1,2] Department of Telecommunications Széchenyi István University Györ, 9026, Hungary

[1,2]E-mail: salam.allawi@sze.hu repas.sandor@sze.hu

[1]Department of Computer Science, College of Computer Science & Information Technology, University of Al-Qadisiyah Al Diwaniyah, 58001, Iraq

[1]E-mail: salam.allawi@qu.edu.iq

that can be used in many fields and datasets. Ongoing research in anomaly detection aims to enhance the sophistication and flexibility of approaches that meet the changing needs of different and complex datasets [8].

Anomaly detection tools, whether they be statistical or machine learning-based, need to be able to identify "point anomalies" (single data observations), "contextual anomalies" (observations that occur in some conditions but not others), and "collective anomalies" (a series of observations) [9], [10].

In recent years, deep learning has improved at finding such complex correlations in high-dimensional data. However, deep learning for anomaly detection has not been sufficiently explored due to the following two significant challenges: (i) it is difficult to obtain large-scale labeled data to train anomaly detectors due to the prohibitive cost of collecting such data in many anomaly detection application domains; and (ii) anomalies often exhibit different anomalous behaviors, making them dissimilar to each other, which poses significant challenges to widened anomaly detection application domains. [11].

Log files record application, system, and network events over time. These files may include application or system problems, warnings, and messages. Log files can indicate flaws and improvement opportunities for technical troubleshooting and debugging. The log files contain valuable information about the status and events of devices, programs, and networks. Log files are good sources for gathering data for debugging and diagnostics or investigating unusual activities [12].

In the present scenario, the application of anomaly detection proves to be advantageous in the identification of performance deficiencies, security breaches, or atypical resource utilization inside the encryption procedures on single-board microcomputers based on the ARM architecture. The utilization of this technology aids in the preservation of the stability and security of these systems by promptly notifying users of any unforeseen actions and facilitating fast interventions [13].

By integrating anomaly detection techniques into the examination of performance, one can acquire useful insights regarding the behavior and performance of Domain Name System 64 (DNS64) applications operating on various open-source operating systems. The process aids in the identification of faults, anomalies, and disparities in performance, so enabling us to conduct informed comparisons and maintain the stability of our DNS64 systems [14].

II. PROBLEM STATEMENT

This part begins by providing a comprehensive explanation of fundamental concepts and terminology of anomaly detection in log files using machine learning techniques.

Cyber Security:

The term "cyber-security" refers to a broad umbrella term encompassing various methods used to keep computers, networks, and data safe from threats such as hackers. Cybersecurity is the practice of ensuring a secure and private digital infrastructure [15]. Experts and professionals in the field of cybersecurity devote a great deal of time and energy to developing a wide variety of cybersecurity systems and technologies, all to protect the privacy, security, and accessibility of sensitive data [16].

Machine Learning:

Machine learning is a form of artificial intelligence that enables computers to learn new tasks without being provided with any instructions. The goal is to build flexible, dynamic computer programs that easily incorporate new data. It falls into two distinct types: supervised and unsupervised. The idea is to build models with the right features to do the right tasks. Classification (both binary and multi-class), clustering (in regression), and modeling (both descriptive and predictive) are all instances of this type of problem [17].

Supervised Learning:

Algorithms that fall under supervised learning can apply prior knowledge to new data if they can access labeled data. Support Vector Machine (SVM), K-Nearest Neighbours algorithm (KNN), Regression, and Random Forest are all examples of algorithms that belong to the machine learning technique; Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are examples of algorithms that belong to the deep learning category

[17]. Algorithms that fall under supervised learning can apply prior knowledge to new data if they can access labeled data. SVM, KNN, Regression, and Random Forest are all examples of algorithms that belong to the machine learning technique; CNN and RNN are examples of algorithms that belong to the deep learning category [17].

Deep Learning:

It is a distinct sub-discipline that falls under the broader umbrella of machine learning. The approach entails the deployment of intelligent networks of numerous layers, wherein each layer independently extracts and processes complex datasets [17].

Unsupervised Learning:

Unsupervised learning is a machine learning approach where the algorithm is responsible for recognizing patterns, relationships, or structures in a dataset without specific instructions in the form of labeled output. Unsupervised learning does not have a predetermined target variable for the algorithm to predict. The system autonomously analyzes the data to reveal underlying structures, groupings, or relationships present in the input information. Unsupervised learning commonly involves clustering, grouping related data points, and dimensionality reduction, simplifying the dataset by capturing its important qualities. Unsupervised learning is used when the patterns are unknown, making it useful for exploring data and uncovering hidden insights [17].

Unsupervised learning involves learning from unlabeled data. Determine the typical outward distance from cluster centers to evaluate data clusters. Unsupervised learning includes association learning and identifying latent variables like film genres. Overfitting is a concern in supervised learning. Assigning each data point its cluster lowers the average distance to the cluster center to zero, but this is not beneficial. Data-driven computations. Key unsupervised learning algorithms are clustering (Hierarchical, K-means) and dimensionality reduction Principal Component Analysis(PCA), t-distributed Stochastic Neighbour Embedding(tSNE), Non-negative Matrix Factorization(NMF) [17], [18] [19]. **Fig. 1** shows a machine learning diagram.

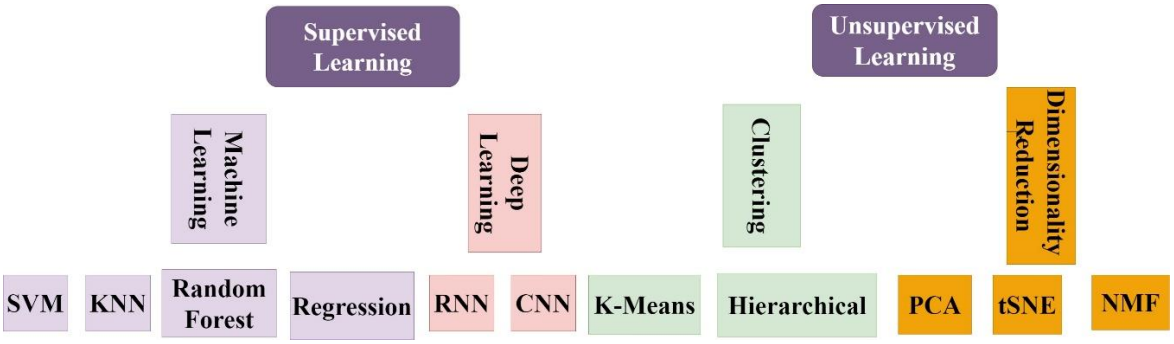

Figure 1. Machine Learning Diagram [19]

ANOMALY DETECTION IN GENERAL:

Anomalies:

Anomalies are unusual events. "Anomaly" comes from the Greek "anomolia," meaning "uneven" or "irregular." An anomaly is something unusual [18].

Statistically Incorrect Predictions:

Parameterized statistical models like the Gaussian distribution are intriguing because parameter estimation is easy, and there are statistical tests that provide well-founded confidence for anomaly identification [20], [21], valid structural assumptions are needed for anomaly detection and model estimate. When fitting a Gaussian model to data from an almost uniform distribution, the probability density at the center (mean) and tails will be exaggerated and underestimated, respectively [22]. Furthermore, incorrect predictions occur when forecasting models or algorithms cannot effectively predict future outcomes based on the available data. The errors may result from incorrect assumptions, insufficient or unbalanced data sources, or limitations in the statistical methods employed. Incorrect information can lead to incorrect decisions that impact several sectors, such as banking, healthcare, and climate

science. Addressing these weaknesses is crucial to improving the reliability and effectiveness of predictive models, enabling more informed decision-making and reducing the potential risks linked to unreliable predictions. [20].

III. RELATED WORK

Analyzing Deep Learning with Log Anomalies:

Using Security Information and Event Management (SIEM) may contain log data consisting of over 1.4 billion logs per day, [23] were able to detect suspicious business-specific activity and user profile behavior. The project encountered difficulties due to scalability issues, noisy data, and no ground truth. The proposed approach requires a feature vector for each internet host using historical data. Unsupervised clustering based on data-specific properties is used to identify potential security issues. Experts in manual labeling should be aware that they are operating in a vacuum. The method is rule-based, and subject-matter expertise is needed to process historical logs. Anomaly detection in log data without prior domain knowledge was proposed by [24]. The suggested method entails a mechanism for finding irregularities in logs and a methodology for diagnosing log key and parameter value deviations. A neural network-based technique is used to forecast the likelihood of the subsequent log key. Similar to how an irregularity in a log parameter sequence can be identified using a Long Short-Term Memory (LSTM) neural network. The software also uses manual feedback on false positives to enhance its accuracy. (LSTM) treats the log series as if it were a sequence of words in a language and treats it as such [25] proposed a deep learning model for detecting log message irregularities [25] using open datasets from BlueGene/L (BGL), Thunderbird, Open Stack, and IMDB. They show that their method may be used for various categorization problems by using the Internet Movie Database (IMDB) dataset. [26] employed Natural Language processing techniques (NLP) to find unusual log entries. A classification LSTM deep learning technique is used to cap off the research using word2vec and TF-IDF feature extraction approaches. They found that for log message identification tasks, word2vec is superior to Term Frequency - Inverse Document Frequency (TF-IDF). In 2019, [27] developed an attention-based LSTM model that could simultaneously spot sequential and computably inexplicable anomalies. Frequent Template Tree (FT-Tree) analyzes logs and creates a new word representation approach called template2vec, which efficiently uses synonyms and antonyms to find abnormalities. This approach addresses the data loss problem when only the log template index is evaluated in [24], and the semantic log connection cannot be provided. Using NetEngine40E [28] analyzed the behavior type, attributes, and rank of routers [8].

Data from the logs were analyzed. Anomaly detection [29], [30], [31], failure diagnosis [32], [33], [34] programmer verification [35], [36], and act prediction [37] are just some of the ways that log analysis has been used to improve software system dependability [38] In recent years, there has been a lot of focus on log parsing and log mining, the two main components of log analysis methods. [5] analyze the performance of four offline log parsing techniques that do not use the system source code: Simple Logfile Clustering Tool.

(SLCT) [39], Iterative Partitioning Log Mining (IPLOM) [23], Message Signature Based Algorithm (LogSig) [40], and Log Key Extraction (LKE) [41]. An offline, linear-time, and space-consuming log parsing approach is proposed in [42], [31] to provide a system-level, real-time approach to analyzing logs [31]. use principal component analysis to detect outliers, with a log-based matrix as input. Using system logs, [35] construct a finite state machine to define the system's behavior at runtime. As mentioned earlier, we focus on log-based anomaly detection techniques instead of the more generalized approaches used in the papers.

Threshold-Based Approaches:

Packet Internet or Inter-Network Groper (Ping) and Hypertext Transfer Protocol (HTTP) detection are the most used network health analysis methods in business. Active detection detects network traffic data anomalies using a specified parameter based on operations and maintenance personnel knowledge and skill. Set a threshold depending on a reasonable server latency range. This detection method is computationally cheap since swing points are hard to locate and close to the normal distribution. It must be more adaptable to different network traffic conditions [43].

Statistical-Based Approaches:

Algorithms based on statistics and probability models use extreme value analysis and hypothesis testing to uncover "anomalies" under assumptions. They might assume a Gaussian distribution for the most basic one-dimensional data and treat outliers as points whose distance exceeds a preset range. After generalizing to high dimensions, one can address each dimension separately and add its anomaly [43].

Methods Based on Distance:

The foundation Distance-representing similarity algorithms identify anomalous points from regular ones [8] suggesting the distance-based proximity detection K-nearest neighbor (KNN) algorithm for Wireless Sensor Network (WSN) anomalous flow data analysis. KNN classifiers identify anomalies. k and n determine each point's k-nearest neighbor distance. We can get the top n outliers by ranking the distances to the k-nearest neighbors. They tested the KNN algorithm using (QualNet) "is a testing and simulation tool offered by Scalable Network Technologies, Inc. It is network simulation software that replicates the actions of a real communications network, serving as a tool for planning, testing, and training". PCA-based anomaly detection is also distance-based. The sample abnormality calculates the weighted Euclidean distance between each sample and the hyperspace formed by k feature vectors [44] studied PCA use scenarios, established a general distance formula, and presented an Internet of Things (IoT) identification strategy using PCA. Several experiments confirmed their strategy. Large datasets make distance-based network traffic anomaly detection algorithms computationally expensive.

Density-Based Approaches:

Local outlier frequency (LOF) assigns a score to each occurrence based on the local density of its neighbors, indicating outliers. High LOF suggests an outlier. Some models have emerged from this core idea. [45] used LOF and Density-based spatial clustering of applications with noise (DBSCAN) to modify settings to change data dynamically. It enhanced network traffic simulation accuracy. Experimental results show the increased LOF algorithm strategy's practical value. Cluster-based local outliers factor (CBLOF) pre-classifies data using K-means clustering before running the LOF algorithm [46], Compute-intensive algorithms like distance-based ones need help with high-dimensional data sets. Density-based anomaly detection algorithms cannot detect fast-changing network traffic anomalies because they require extensive parameter selection knowledge.

Neural Network-Based Approaches:

Anomaly detection in network traffic requires massive, sophisticated datasets. Neural network techniques excel in complex contexts [47]. Kim and Cho [48] developed the Convolutional Long Short-Term Memory method (C-LSTM) to extract more complex features using a convolutional neural network (CNN), a long short-term memory (LSTM), and a deep neural network (DNN). It outperformed other state-of-the-art machine learning techniques. Wei and Wang [49] used a CNN and an RNN to create a hierarchical spatiotemporal feature learning (HAST-NAD) network anomaly detection approach. The time series characteristics methodology outperforms the spatial characteristics algorithm. Deep learning works well but requires lots of high-quality tagged data.

Autoencoder:

The use of deep learning methods for anomaly identification has increased dramatically in recent years [1], [50]. The autoencoder is a method widely used in deep learning for anomaly identification. An autoencoder is taught to recreate the input data, and the difference between the two is used to quantify anomalies.

Isolation Forest:

Outliers can also be found using the Isolation Forest approach (IForest). This tree-structured algorithm picks a feature randomly and then a split value between its extremes [51], [52] The algorithm splits data repeatedly and utilizes the number of splits needed to isolate a point as a measure of irregularity.

Isolation Forests are ideal for network traffic anomaly detection because they can handle huge data volumes, high-dimensional data, and low abnormality fractions. Even with massive datasets, iForest beats rival methods. Forrest is an ensemble learning method that learns complex patterns using isolation trees. A simple discriminator is each tree. Each weak classifier's training set is a bootstrap sample from the entire dataset. iForest achieves good generalization performance and avoids overfitting by integrating all isolation tree outputs. The hyperplanes utilized in Hariri et al.'s extended isolated forest method (EiForest) [53] are not parallel to the coordinate frame. EiForest accounts for bias in the standard isolated forest example that could skew outlier results. EiForest is stronger in experiments. The EiForest removes bias but does not account for the anomalous ratio's effect on the inquiry, [54] introduced a sliding window frame-based iForest-based adaptive streaming data anomaly detection method for computer and sensor network traffic data. Isolated forests and reduced-dimensional variable selection were

suggested for anomaly detection [55]. Due to optical emission spectral data's high dimensionality and correlation, anomaly detection performance is low. This method is more interpretable than PCA.

Detecting Conformal Anomalies (Offline Vs. Online):

Offline Anomaly Detection:

The offline anomaly detection technique processes all log data at once. This approach processes massive amounts of historical log data or analyzes logs periodically. For offline log file anomalies, the statistical approach is utilized. Log data abnormalities and patterns are searched for [56]. Statisticians follow these steps:

Log data is cleaned by removing metadata like timestamps.

Log data features include message frequency, inter-message interval, and message size.

Utilizing features, a model of the system's typical behavior is created. Classification, clustering, and time-series analysis can help.

Log data abnormalities. Anomaly detection compares it to a model of typical behavior. Z-scores, chi-squares, and distance-based methods are statistical tools.

False positives and severity rankings are removed from anomalies.

Since it can analyze the entire dataset, the offline method may be more precise and dependable than real-time or streaming algorithms. It may be computationally expensive for real-time anomaly detection, especially for large datasets [56].

To conclude, the offline log file anomaly identification technique uses statistical methods to imitate the system's normal behavior and find any anomalies. This method may be computationally costly and better suited for real-time anomaly detection. Only batch processing of historical log data or systematic study of logs is proper [56].

Online Anomaly Detection:

Online anomaly detection systems find anomalies in log data using statistical approaches. Machine learning methods, such as clustering and classification, find anomalies. Combining these methods with heuristics or rules-based methods improves anomaly detection.

Online anomaly detection systems can adjust to new and changing data. As data is uploaded, the algorithm's model is improved to detect new anomalies. Online algorithms can more easily detect emerging dangers [56].

Online algorithms also have speed and scalability advantages. Real-time anomaly detection can significantly response reduce security incident investigation and mitigation time. Online algorithms process large volumes of data without pre-processing or sampling [56]. Thus, online anomaly detection techniques in log files give computer systems a valuable and efficient tool for harmful activity detection. These algorithms continuously analyze log data for abnormalities, improving security and reducing data breaches [56].

IV. METHODOLOGY

Logs:

Unstructured information is frequently printed chronologically in logs generated by large data systems. Each line in a log file often contains a constant and a variable. The source code statements that output constant messages are the constant component. These constants are the building blocks from which log keys are derived; log keys are the messages consistently present across a set of log entries. The following is an example of a web server log file [18] [57]. Fig. 2 shows A sample of the web server Log File.

```
216.239.46.60 - - [04/Jan/2003:14:56:50 +0200] "GET
/~lpis/curriculum/C+Unix/Ergastiria/Week-7/filetype.c.txt HTTP/1.0"
304 -
216.239.46.100 - - [04/Jan/2003:14:57:33 +0200] "GET
/~oswinds/top.html HTTP/1.0" 200 869
64.68.82.70 - - [04/Jan/2003:14:58:25 +0200] "GET /~lpis/systems/r-
device/r_device_examples.html HTTP/1.0" 200 16792
216.239.46.133 - - [04/Jan/2003:14:58:27 +0200] "GET
/~lpis/publications/crc-chapter1.html HTTP/1.0" 304 -
209.237.238.161 - - [04/Jan/2003:14:59:11 +0200] "GET /robots.txt
HTTP/1.0" 404 276
209.237.238.161 - - [04/Jan/2003:14:59:12 +0200] "GET
/teachers/pitas1.html HTTP/1.0" 404 286
216.239.46.43 - - [04/Jan/2003:14:59:45 +0200] "GET
/~oswinds/publications.html HTTP/1.0" 200 48966
```

Figure 2. A sample of Web Server Log File [18] [57]

In **Fig. 2**, you can see that a timestamp precedes each entry in the log and then either an INFO or DEBUG event type. Here you may see what kind of log entry this event will make. After an event occurs, a report detailing the occurrence and a log of the commands and steps taken during testing are generated.

Log-Parsing:

Log file parsing analyzes log files to get insight into a computer system or network. Log files include valuable data that can be used to understand better a system's operation, security, and user behavior [18]. A log file must be parsed into individual log messages before any useful information can be retrieved from the file. Metadata includes information such as the time and location of a message's transmission, the sender's identity, and the message's content. Programmers and system administrators can better comprehend how a system works and identify potential bottlenecks or vulnerabilities using this information [18].

Drain:

Drain, a web-based log parser, processes multiple logs quickly. A depth-defined parse tree stores parsing rules. Most log parsing methods batch-process logs offline. As long volumes rise, offline log parsing model training takes time. Drain directs log group search with a fixed-depth tree to avoid an intense and imbalanced tree [18] [8].

Log parsing helps organize unstructured log messages. Drain initially analyzes raw log messages using domain knowledge and simple regular expressions. Then, the internal log's rules search the log group (the tree's leaf node). **Fig. 3** shows the framework of anomaly detection example.

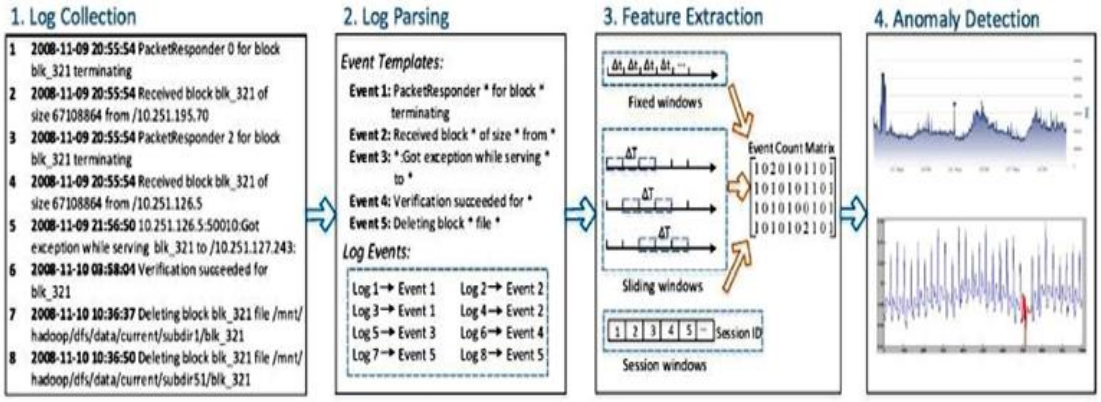

Figure 3. Framework of Anomaly Detection Example [8]

V. DATASETS & EVALUATION

Datasets:

This section is with common datasets used. the table below illustrates these datasets. **Fig. 4** shows. % of the dataset used.

Table 1. Public datasets used [8]

No.	*Dataset*	*No. of Studies*
1	NSL-KDD	2
2	KDD-Cup99	8
3	ISCX	1
4	ECMLPKDD2007	1
5	UNSW-NB15	2
6	Kyoto2006	2
7	DARPA	4
8	CSIC	13

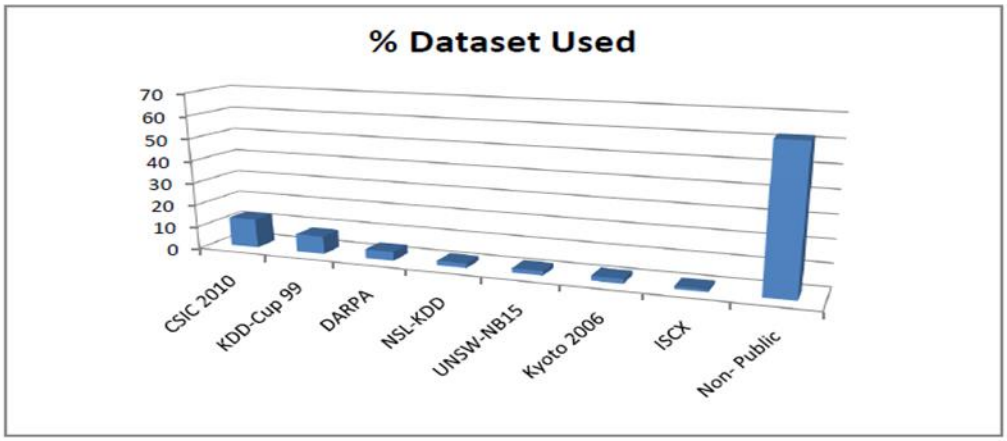

Figure 4. % of Dataset used [8]

Evolution:

This section begins with a comprehensive explanation of the basic concepts and terminology of Accuracy, Precision, Recall, and F1 Score.

Accuracy:

Accuracy is the most straightforward performance indicator as a percentage of accurately anticipated observations to the total number of observations. The model with the highest accuracy is best [58].

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (1)$$

Precision:

Precision measures how many expected positive observations were positive. A low false positive rate is synonymous with high precision [58].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall:

Correctly predicted positive observations as a percentage of all observations in the actual class; this is what we mean by recall [58]. The range of its values is from 0 to 1. The better the model performs, the higher the recall.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F1 Score:

The F1 Score considers both accuracy and recall for a final tally. As a result, the likelihood of a false positive or false negative is calculated [58]. The greater the F1 score, the better the models $F1 - Score = \frac{2*Pecision*Recall}{Precision+Recall}$ (4)

Cross-prediction logging prediction strategies were studied in this paper [59]. EC-Logger, an innovative catch-block logging prediction model, was proposed. Ensemble approaches merged nine base classifiers. Three open-source Java projects—Tomcat, Cloud Stack, and Hadoop—were utilized to assess EC-Logger's efficacy. Bagging, average, and majority vote make this classifier better than the gold standard. EC-Logger Average Vote is the best. Cloud Stack outperforms competitors in transferability. So this paper is relevant to the research problem by testing and selecting the optimal machine learning algorithm. They chose the best model using an ensemble method. [59].

LogOpt predicts automated catch block logging [60]. LogOpt uses code-based static properties to train models. The code base yielded 46 logging prediction traits. LogOpt was tested on Apache Tomcat and CloudStack, with results. In the paper [61], the authors propose DeepLog, a model of a deep neural network that uses Long Short-Term Memory (LSTM) to represent a system log in terms of a sequence of words.

This enables DeepLog's anomaly detection capabilities. It can automatically learn log patterns from regular operations and identify outliers when log patterns differ from the model learned from log data under normal conditions. This thesis work will attempt to deploy and explore the performance of other machine learning techniques, which would set it apart from prior related works [61], [59].

Online anomaly detection using time-series visualizations of security indicators and predictive models [62]. This clustering method connects log line clusters using static cluster maps to identify cluster transitions. Their strategy focuses on loglines with unusual occurrence rates, periodicities, and correlations. They distinguished between temporal anomalies, which modify system behavior over time, and highly distinct lines, which occur once as outliers. This approach learns without prior knowledge of assaults or logs data types.

In [63], the goal is first to identify normal log file profiles and then detect any abnormal behavior. The team developed a lightning-quick technique for extracting line patterns from unprocessed log files. The method used the characteristics of log files as input. They have decided to use an algorithm for clustering data.

The log's most frequent terms are used to create a table of cluster options. A line can be formed by two or more commonly used words. The algorithm clusters candidates with count values above the cut-off. Using term-weighting approaches from information retrieval, Principal Component Analysis (PCA) gave the best anomaly detection results on both feature matrices with minimal parameter change. Principal component analysis (PCA) finds statistically dominant patterns and irregularities in data, including frequent pattern mining.

The k-means approach was used to identify highly coherent clusters of anomalous and normal occurrences in the paper [64]. To create an easily understandable set of rules from the previously binary clustered data, XGBoost was built as a gradient tree-boosting technique. The guidelines provided the foundation for expanding its use to a wide range of previously unforeseen phenomena in a decentralized computing setting.

This allowed them to acquire categorized instances of abnormality.

Data mining methods such as the K Means clustering algorithm and the RBF kernel function of the Support Vector Machine's classification module are combined in the hybrid strategy suggested in the article [65]. The primary goal of their method is to reduce the data characteristics for each observation. When applied to the KDDCUP'99 Data Set, the proposed method resulted in a higher Detection Rate and accuracy.

The study [66] developed an Intrusion Detection System (IDS) to detect Denial of Service (DoS) attacks in IoT networks using machine learning algorithms like Decision Trees, Support Vector Machines, Random Forests, and k-nearest Neighbours. The data was divided into training and testing subsets, with 67% used for training and 33% for testing. The performance of the classification algorithms was evaluated using metrics like accuracy, precision, recall, and F1 score. The study highlighted the importance of feature selection in enhancing IDS accuracy and reliability [66].

The study [67] aims to improve anomaly detection in industrial settings by training a neural network to focus on specific areas of interest. The authors introduce a novel loss function called Cross-Entropy Overlap Distance (CEOD) to reduce false positives caused by background noise defects. The approach outperforms standard cross-entropy loss on both benchmark and real-case industrial datasets. The study used three different feature selection approaches: all features, Genetic Algorithm (GA)-selected features, and Correlation-based Feature Selection (CFS). The results showed that DT and RF algorithms performed best when trained with GA features.

The study [68] evaluates a Split Active Learning Anomaly Detector (SALAD) method for network data stream anomaly detection, aiming to reduce labeling costs, improve detection accuracy, and adapt to changes in data streams. Results show high accuracy and F1 scores, making SALAD a valuable approach for real-time intrusion detection systems.

Furthermore, the research shows a transition from conventional approaches to sophisticated and varied machine learning methods, aiming to improve the effectiveness of log file analysis for anomaly identification. Integrating deep learning, ensemble approaches, and hybrid approaches indicates an advancement in this area, emphasizing practical application and performance enhancement.

VI. CONCLUSION AND FUTURE DIRECTION

This paper examined the limitations and difficulties of current anomaly detection techniques, such as identifying uncommon anomalies, coping with complex log patterns, and interpreting results. These difficulties underscore the need for additional research and development in anomaly detection.

There are several promising prospective research directions for anomaly detection in log files. Integrating multimodal data sources to improve anomaly detection is one method of exploration. By combining various categories of data, such as logs, network traffic, and system metrics, it is possible to gain a deeper understanding and improve the effectiveness of abnormal behavior. In the future, we will use an autoencoder as a feature selection, a convolutional neural network (CNN) algorithm, an Isolation Forest and Autoencoder algorithm, or Gaussian and Deep Autoencoder Mixture as assault classification predictors considered predictors for attack detection.

REFERENCES

[1] M. Almansoori and M. Telek, “Anomaly Detection using a combination of Autoencoder and Isolation Forest,” no. March, pp. 25–30, 2023, https://doi: 10.3311/wins2023-005.

[2] M. Ahmed, A. Naser Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” J. Netw. Comput. Appl., vol. 60, pp. 19–31, 2016, https://doi.org/10.1016/j.jnca.2015.11.016.

[3] F. Skopik, M. Wurzenberger, M. Landauer, F. Skopik, M. Wurzenberger, and M. Landauer, “Survey on log clustering approaches,” Smart Log Data Anal. Tech. Adv. Secure. Anal., pp. 13–41, 2021. http://dx.doi.org/10.1016/j.cose.2020.101739.

[4] M. Ahmed et al., “A survey of anomaly detection techniques in financial domain,” Futur. Gener. Comput. Syst., vol. 55, no. 1, pp. 19–31, 2016, https://doi: 10.1016/j.jnca.2015.11.016.

[5] M. Ahmed, A. Naser Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” J. Netw. Comput. Appl., vol. 60, pp. 19–31, 2016, https://doi: 10.1016/j.jnca.2015.11.016.

[6] D. M. Hawkins, “Identification of Outliers,” Identif. Outliers, 1980, https://doi: 10.1007/978-94-015-3994-4/COVER.

[7] J. Patterson and A. Gibson, Deep learning: A practitioner’s approach. “O’Reilly Media, Inc.,” 2017. https://www.oreilly.com/library/view/deep-learning/9781491924570/

[8] M. Siwach and S. Mann, “Anomaly Detection for Web Log-based Data: A Survey,” 2022 IEEE Delhi Sect. Conf. DELCON 2022, vol. 13, no. 1, pp. 129–148, 2022, https://doi: 10.1109/DELCON54057.2022.9753130.

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, 2009. https://DOI:10.1145/1541880.1541882

[10] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” arXiv Prepr. arXiv1901.03407, 2019. https://doi.org/10.48550/arXiv.1901.03407

[11] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015. https://www.nature.com/articles/nature14539

[12] P. Nimbalkar, V. Mulwad, N. Puranik, A. Joshi, and T. Finin, “Semantic interpretation of structured log files,” Proc. - 2016 IEEE 17th Int. Conf. Inf. Reuse Integer. IRI 2016, pp. 549–555, 2016,https://doi: 10.1109/IRI.2016.81.

[13] S. R. Repas, “Performance Analysis of Encryption Capabilities of ARM-based Single Board Microcomputers,” Infocommunications J., vol. 15, no. 2, pp. 8–13, 2023, https://doi: 10.36244/ICJ.2023.2.6.

[14] G. Lencse and S. Répás, “Performance analysis and comparison of four DNS64 implementations under different free operating systems,” Telecommun. Syst., vol. 63, no. 4, pp. 557–577, 2016, https://doi: 10.1007/s11235-016-0142-x.

[15] F. Ullah and M. A. Babar, “Architectural tactics for big data cybersecurity analytics systems: a review,” J. Syst. Softw., vol. 151, pp. 81–118, 2019. https://doi.org/10.1016/j.jss.2019.01.051

[16] S. Dua and X. Du, Data mining and machine learning in cybersecurity. CRC Press, 2016.

[17] A. Helwan and D. Uzun Ozsahin, “Sliding Window Based Machine Learning System for the Left Ventricle Localization in MR Cardiac Images,” Appl. Comput. Intell. Soft Comput., vol. 2017, 2017, https://doi: 10.1155/2017/3048181.

[18] Z. Zamanian, “Anomaly Detection in System Log Files Using Machine Learning,” no. February, pp. 1–92, 2019. https://studentsrepo.um.edu.my/10748/2/Zahedeh__%E2%80%93_Dissertation.pdf

[19] M. K. Hooshmand and D. Hosahalli, “Network anomaly detection using deep learning techniques,” CAAI Trans. Intell. Technol., vol. 7, no. 2, pp. 228–243, 2022. https://doi.org/10.1049/cit2.12078

[20] R. Laxhammar and G. Falkman, “Online learning and sequential anomaly detection in trajectories,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 6, pp. 1158–1173, 2014, https://doi: 10.1109/TPAMI.2013.172.

[21] “Anomaly detection: A survey: ACM Computing Surveys: Vol 41, No 3.” https://dl.acm.org/doi/10.1145/1541880.1541882 (accessed May 06, 2023).

[22] Y. Shi, C. Long, X. Yang, and M. Deng, “Abnormal Ship Behavior Detection Based on AIS Data,” Appl. Sci., vol. 12, no. 9, 2022, https://doi: 10.3390/app12094635.

[23] P. Pudil and J. Novovičová, “Novel Methods for Feature Subset Selection concerning Problem Knowledge,” Feature. Extr. Constr. Sel., pp. 101–116, 1998, https://doi: 10.1007/978-1-4615-5725-8_7.

[24] V. Jyothsna, V. V. Rama Prasad, and K. Munivara Prasad, “A Review of Anomaly-based Intrusion Detection Systems,” Int. J. Comput. Appl., vol. 28, no. 7, pp. 26–35, 2011, https://doi: 10.5120/3399-4730.

[25] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” J. Syst. Softw., vol. 80, no. 4, pp. 571–583, 2007, doi: https://doi.org/10.1016/j.jss.2006.07.009.

[26] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, “Intrusion detection system: A comprehensive review,” J. Netw. Comput. Appl., vol. 36, no. 1, pp. 16–24, 2013, https://doi: 10.1016/j.jnca.2012.09.004.

[27] R. Samrin and D. Vasumathi, “Review on anomaly based network intrusion detection system,” Int. Conf. Electr. Electron. Commun. Comput. Technol. Optim. Tech. ICEECCOT 2017, vol. 2018-Janua, pp. 141–147, 2018, https://doi:10.1109/ICEECCOT.2017.8284655

[28] M. Kakavand, N. Mustapha, A. Mustapha, M. T. Abdullah, and H. Riahi, “A survey of anomaly detection using data mining methods for hypertext transfer protocol web services,” J. Comput. Sci., vol. 11, no. 1, pp. 89–97, 2015, https://doi: 10.3844/jcssp.2015.89.97.

[29] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino Júnior, “An intrusion detection and prevention system in cloud computing: A systematic review,” J. Netw. Comput. Appl., vol. 36, no. 1, pp. 25–41, 2013, https://doi: 10.1016/j.jnca.2012.08.007.

[30] K. P. Singh, N. Basant, and S. Gupta, “Support vector machines in water quality management,” Anal. Chim. Acta, vol. 703, no. 2, pp. 152–162, 2011, https://doi: 10.1016/j.aca.2011.07.027.

[31] Y. Luo, S. Cheng, C. Liu, and F. Jiang, “PU Learning in Payload-based Web Anomaly Detection,” in 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), 2018, pp. 1–5. https://doi: 10.1109/SSIC.2018.8556662.

[32] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, “A survey of deep learning-based network anomaly detection,” Cluster Comput., vol. 22, pp. 949–961, 2019, https://doi: 10.1007/s10586-017-1117-8.

[33] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems, and challenges,” Comput. Secure., vol. 28, no. 1, pp. 18–28, 2009, doi: https://doi.org/10.1016/j.cose.2008.08.003.

[34] C. Wang, T. T. N. Miu, X. Luo, and J. Wang, “SkyShield: A sketch-based defense system against application layer DDoS attacks,” IEEE Trans. Inf. Forensics Secur., vol. 13, no. 3, pp. 559–573, 2018, https://doi: 10.1109/TIFS.2017.2758754.

[35] A. Y. Bedikian, J. Stroehlein, J. Korinek, D. Karlin, M. Valdivieso, and G. P. Bodey, “Phase II evaluation of dihydroxyanthracenedione (DHAD, NSC 301739) in patients with metastatic colorectal cancer,” Am. J. Clin. Oncol. Cancer Clin. Trials, vol. 6, no. 1, pp. 45–48, 1983, https://doi: 10.1097/00000421-198302000-00007.

[36] R. Xiao, J. Su, X. Du, J. Jiang, X. Lin, and L. Lin, “SFAD: Toward effective anomaly detection based on session feature similarity,” Knowledge-Based Syst., vol. 165, pp. 149–156, 2019, doi: https://doi.org/10.1016/j.knosys.2018.11.026.

[37] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, “A survey of deep learning-based network anomaly detection,” Cluster Comput., vol. 22, pp. 949–961, 2017. https://link.springer.com/article/10.1007/s10586-017-1117-8

[38] M. Zolotukhin, T. Hämäläinen, T. Kokkonen, and J. Siltanen, “Increasing web service availability by detecting application-layer DDoS attacks in encrypted traffic,” in 2016 23rd International Conference on Telecommunications (ICT), 2016, pp. 1–6. https://doi: 10.1109/ICT.2016.7500408.

[39] G. Yuan, B. Li, Y. Yao, and S. Zhang, “A deep learning enabled subspace spectral ensemble clustering approach for web anomaly detection,” in 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 3896–3903. https://doi: 10.1109/IJCNN.2017.7966347.

[40] L. Wang, S. Cao, L. Wan, and F. Wang, “Web Anomaly Detection Based on Frequent Closed Episode Rules,” in 2017 IEEE Trustcom/BigDataSE/ICESS, 2017, pp. 967–972. https://doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.338.

[41] T. Brugger, “KDD cup’99 datasets (network intrusion) considered harmful, 15 September 2007. Retrieved January 26, 2008.” 2007. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5

[42] M. M. Najafabadi, T. M. Khoshgoftaar, C. Calvert, and C. Kemp, “User Behavior Anomaly Detection for Application Layer DDoS Attacks,” in 2017 IEEE International Conference on Information Reuse and Integration (IRI), 2017, pp. 154–161. https://doi: 10.1109/IRI.2017.44.

[43] T. H. Shin and S. H. Kim, “Utility Analysis about Log Data Anomaly Detection Based on Federated Learning,” Appl. Sci., vol. 13, no. 7, 2023, https://doi: 10.3390/app13074495.

[44] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets, and challenges,” Cybersecurity, vol. 2, no. 1, 2019, https://doi: 10.1186/s42400-019-0038-7.

[45] Z. Gan and X. Zhou, “Abnormal Network Traffic Detection Based on Improved LOF Algorithm,” in 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2018, vol. 01, pp. 142–145. https://doi: 10.1109/IHMSC.2018.00040.

[46] Z. He, X. Xu, and S. Deng, “Discovering cluster-based local outliers,” Pattern Recognit. Lett., vol. 24, no. 9, pp. 1641–1650, 2003, doi: https://doi.org/10.1016/S01678655(03)00003-5.

[47] M. Abbasi, A. Shahraki, and A. Taherkordi, “Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey,” Comput. Commun., vol. 170, pp. 19–41, 2021, doi: https://doi.org/10.1016/j.comcom.2021.01.02.

[48] T.-Y. Kim and S.-B. Cho, “Web traffic anomaly detection using C-LSTM neural networks,” Expert Syst. Appl., vol. 106, pp. 66–76, 2018, doi: https://doi.org/10.1016/j.eswa.2018.04.004.

[49] G. Wei and Z. Wang, “Adoption and realization of deep learning in network traffic anomaly detection device design,” Soft Comput., vol. 25, no. 2, pp. 1147–1158, 2021, https://doi: 10.1007/s00500-020-05210-1.

[50] A. Patcha and J.-M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” Comput. Networks, vol. 51, no. 12, pp. 3448–3470, 2007, doi: https://doi.org/10.1016/j.comnet.2007.02.001.

[51] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation-Based Anomaly Detection,” ACM Trans. Knowl. Discov. from Data, vol. 6, no. 1, Mar. 2012, https://doi: 10.1145/2133360.2133363.

[52] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422. https://doi: 10.1109/ICDM.2008.17.

[53] S. Hariri, M. C. Kind, and R. J. Brunner, “Extended Isolation Forest,” IEEE Trans. Knowl. Data Eng., vol. 33, no. 4, pp. 1479–1489, 2021, https://doi: 10.1109/TKDE.2019.2947676.

[54] Z. Ding and M. Fei, “An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window,” IFAC Proc. Vol., vol. 46, no. 20, pp. 12–17, 2013, doi: https://doi.org/10.3182/20130902-3-CN-3020.00044.

[55] L. Puggini and S. McLoone, “An enhanced variable selection and Isolation forest-based methodology for anomaly detection with OES data,” Eng. Appl. Artif. Intell., vol. 67, pp. 126–135, 2018, doi: https://doi.org/10.1016/j.engappai.2017.09.021.

[56] C. Piciarelli and G. L. Foresti, “On-line trajectory clustering for anomalous events detection,” Pattern Recognit. Lett., vol. 27, no. 15, pp. 1835–1842, 2006, https://doi: 10.1016/j.patrec.2006.02.004.

[57] A. Vakali, J. Pokomý, and T. Dalamagas, “An overview of web data clustering practices,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3268, no. May, pp. 597–606, 2004, https://doi: 10.1007/978-3-540-30192-9_59.

[58] “(6) Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures | LinkedIn.” https://www.linkedin.com/pulse/accuracy-precision-recall-f1-score-interpretation-mukul-choudhary/ (accessed May 07, 2023).

[59] S. Lal, N. Sardana, and A. Sureka, “ECLogger: Cross-project catch-block logging prediction using an ensemble of classifiers,” E-Informatica Softw. Eng. J., vol. 11, no. 1, pp. 7–38, 2017, https://doi: 10.5277/e-Inf170101.

[60] S. Lal and A. Sureka, “LogOpt: Static Feature Extraction from Source Code for Automated Catch Block Logging Prediction,” 2016, pp. 151–155. https://doi: 10.1145/2856636.2856637.

[61] E. Haas, “Beitrag zum Morbus Hodgkin und seiner Studieneinteilung,” Klin. Wochenschr., vol. 31, no. 29–30, pp. 694–697, 1953, https://doi: 10.1007/BF01473650.

[62] M. Landauer, M. Wurzenberger, F. Skopik, G. Settanni, and P. Filzmoser, “Dynamic log file analysis: An unsupervised cluster evolution approach for anomaly detection,” Comput. Secur., vol. 79, pp. 94–116, 2018, doi https://doi.org/10.1016/j.cose.2018.08.009.

[63] X. Cheng and R. Wang, “Communication network anomaly detection based on log file analysis,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8818, pp. 240–248. https://doi: 10.1007/978-3-319-11740-9_23.

[64] J. Henriques, F. Caldeira, T. Cruz, and P. Simões, “Combining k-means and xgboost models for anomaly detection using log datasets,” Electron., vol. 9, no. 7, pp. 1–17, 2020, https://doi: 10.3390/electronics9071164.

[65] U. Ravale, N. Marathe, and P. Padiya, “Feature selection based hybrid anomaly intrusion detection system using K Means and RBF kernel function,” Procedia Comput. Sci., vol. 45, no. C, pp. 428–435, 2015, https://doi: 10.1016/j.procs.2015.03.174.

[66] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, “Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms,” Sensors (Basel)., vol. 24, no. 2, 2024, https://doi: 10.3390/s24020713.

[67] M. Fraccaroli, A. Bizzarri, P. Casellati, and E. Lamma, “Exploiting CNN’s visual explanations to drive anomaly detection,” Appl. Intell., vol. 54, no. 1, pp. 414–427, 2024, https://doi: 10.1007/s10489-023-05177-0.

[68] C. Nixon, M. Sedky, J. Champion, and M. Hassan, “SALAD: A split active learning based unsupervised network data stream anomaly detection method using autoencoders,” Expert Syst. Appl., vol. 248, no. November 2023, p. 123439, 2024, https://doi: 10.1016/j.eswa.2024.123439.