Software Requirement Specification Document for Eyes On Insiders: Log Anomaly Detection Of Insider's Abnormal Behaviours Using Machine Learning

Mohand khaled,Yasmine Mostafa, Jana Ahmed , Hoda Amr
Supervised by: Assoc. Prof. Diaa Salama ,Eng. Salma Osama

April 9, 2024

Table 1: Document version history

Version	Date	Reason for Change
1.0	14-Jan-2024	SRS First version's specifications are defined.
1.2	9-March-2024	Overview and UML edited
1.3	1-April-2024	Detailed Functional Specification Updated

GitHub: https://github.com/hudaamr/eyesoninsider.git

Contents

Abstract

Recently, organisations have been focusing just on external attacks, neglecting the threat posed by insiders. Insider threats are now one of the most massive and damaging threats that could happen for any system. Also, the log files are the target for any insider attack. Therefor, This project proposes a secure machine learning system for log insiders attacks to determine the normal and abnormal behaviours after scanning the log files, and then send warning when detecting any irregular action. So, the paper introduces a desktop application that highlights the idea of securing systems from internal malicious doubts using some machine learning algorithms like(SVM-Isolated forest-Kmeans). Those algorithms briefly collects raw logs, parsing those logs, then extract features and finally detect any danger with high effectiveness in evaluation matrix stage. At last, the impact that the paper aims is to focus on the idea of implementing an applicable and essential detector application to stop precariousness.

1 Introduction

1.1 Purpose of this document

The goal of the eyes on insiders SRS document is to illustrate detailed documentation of eyes on insiders project. Eyes on insiders is a desktop application used to detect insider threat by collecting logs, extract features and detect anomalous users which may contain potential insider threats. Firstly, the purpose and scope of this document will be provided. Moreover, system functionality such as classification techniques are utilized to learn a model to maximize the discrimination between normal and abnormal instances. Finally, the used datasets 'Final IP mapped data' and 'Loghub', an AI modular object detection library.

1.2 Scope of this document

The document's scope is to outline the overview of the project and the process of log analysis for anomaly detection with three main classifiers (i.e. Isolation Forest, SVM, and K-Means).Furthermore, this document goes through the functional and non-functional requirements of the eyes on insiders desktop application and system, the software and hardware limitations, the data design, and the object-oriented class diagram. Finally, this document also covers the operational scenarios of the system and the exact timeline of how this desktop application will be developed.

1.3 Business Context

Businesses bolster their networks against outside malicious attacks by investing in security defenses. They fail to implement security against potential threats posed by malicious or compromised insiders. However, Perpetrators may exploit their authorized entry to vital systems to ultimately pilfer or alter data systems for financial gain or malignant purposes. As the usage of information and communication technologies in the business world increase, so do the opportunities for employees to steal information against the companies they work for. According to a survey cited in Gurucul's 2020 Insider Threat Report, approximately 82% of security practitioners believe their organization's insider threat efficacy is "somewhat effective," "very effective," or "extremely effective." [1] .

2 Similar Systems

2.1 Academic

The researchers provide a system in [2], concentrating on log-based anomaly detection while utilising online learning and robust feature extraction. The main problem that is discussed is the constraints that conventional machine learning algorithms have when it comes to anomaly detection. These limits include high false alarm rates, extended training times, and the need for labeled data. In order to address these issues, the researchers suggest a framework **Figure 1** robust feature extraction to reduce noise and online evolving anomaly detection to change parameters dynamically. They use the Online Evolving SVM algorithm as an example of an online anomaly detection technique to demonstrate this approach. System logs from a sizable computing cluster were the source of a publicly accessible dataset that the researchers used for their investigation. Results indicate that, regarding accuracy and efficiency, their system outperforms the most advanced anomaly detection algorithms available today. Given that the paper focuses specifically on system logs from a large-scale computing cluster, one possible criticism of it is the representativeness of the dataset. To address this, the researchers recognize that additional testing on a variety of datasets is required to determine the efficacy of the framework in a range of settings.

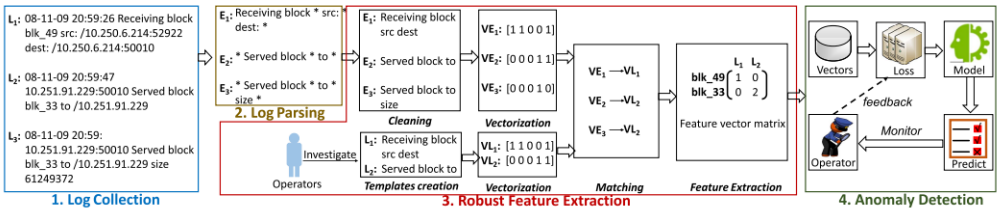

Figure 1: ROEAD framework
[2]

Zibin Zheng[3] delivers a comprehensive assessment of the research on automated log parsing with the goal of addressing the problems related to manual log observation. To turn log messages into structured data for additional analysis, automatic log parsing is necessary because it is impractical to manually scan unstructured logs. In this respect, the researchers provided a contribution by evaluating 13 log parsers on 16 log datasets spanning various systems and applications. A review was conducted on the log parsers' correctness, robustness, and efficiency; These elements are essential for automated log parsing applications in the real world. The researchers additionally wrote about the results they had achieved and the conclusions they obtained from an industrial application at Huawei. Distributed systems, supercomputers, operating systems, mobile systems, server apps, and standalone software were among the datasets that the researchers used. The study's main outcomes demonstrated that different parsing methods and modes had various impacts on the accuracy and efficiency of log parsers. Significant suggestions have been issued by the researchers for additional research and the implementation of automated log parsing. One possible critique of the paper, though, is that the analysis study only used a limited number of datasets and log parsers, so they may not have been completely representative of all scenarios.

A comprehensive review of the latest advancements in deep anomaly detection. The main problem statement of [4] is to address the unique complexities and challenges that arise in anomaly detection, such as the need to detect anomalies from multiple heterogeneous data sources and the incorporation of conditional/group anomalies into anomaly measures/models. To solve this problem, the researchers formulated the state-of-the-art deep anomaly detection techniques into three guiding frameworks: end-to-end anomaly score learning, learning representations of normalcy, and deep learning for generic feature extraction. **Figure 2**

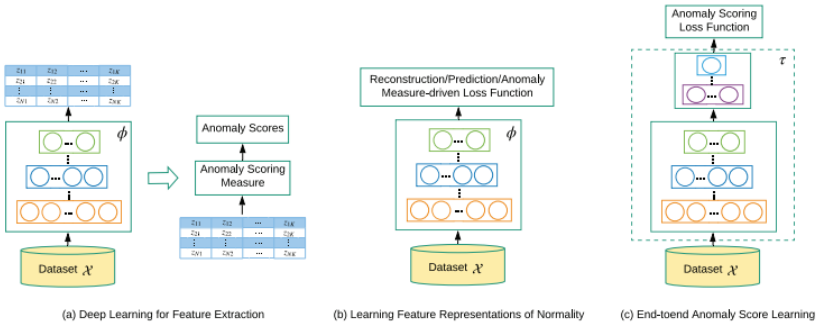

Figure 2: three principled frameworks
[4]

They also presented a taxonomy in a hierarchical structure to group the techniques according to 11 distinct modeling perspectives and conducted a comprehensive literature review of relevant studies in leading conferences and journals. The researchers did not use a specific dataset for this review, as their focus was on the methodology and techniques used in deep anomaly detection. The main results of the review include a better understanding of the key intuitions, objective functions, and underlying assumptions of deep anomaly detection methods, as well as possible future opportuni-

ties and new perspectives for addressing the challenges in anomaly detection using deep learning. The absence of a particular case study or real-world application of the techniques being reviewed, Nevertheless, is a valid criticism of the work. The results' potential use in real-world situations may be limited by this omission.

In[5],Deep learning neural networks are being used by the researchers to create an automated monitoring system for the IBM Cloud Platform. The work's primary issue is the proliferation of false alarms in the current monitoring and alerting system as a result of the high dimensionality and non-Utilizing statistical models is challenging due to the data's linearity. To address this issue, the investigators designed a microservice-based, layered architecture that builds a dependable, salable data collection pipeline.The pipeline has the ability to draw and collect data from multiple sources, such as databases and message queues.After the coll-After being verified and standardized, collected data are routed to the analytics section, which designates observations as typical or unusual.The Console DevOps team is notified of the anomalies for review and confirmation. The processing pipeline is made to be independent of the type of data that is gathered, and different software, hardware, and network metrics can all be evaluated and analyzed using the same solution. To train the deep learning neural networks, the researchers used a dataset of telemetry collected in close to real-time from hundreds of components. The proposed monitoring system was able to detect anomalies with high precision and record them up to 20 minutes earlier than the previous one, which was the main outcome of the researchers' effort. However, the study did not offer a thorough examination of the restrictions and potential negative effects of the suggested remedy, which would have been beneficial for further studies in this field.

2.2 Business Applications

2.2.1 Splunk

A platform that helps people understand and use lots of information.It collects and looks at data in real-time, allowing users to quickly find and fix problems.It also warns users about potential issues and keeps track of security and rules.[6]

Figure 3: splunk
[6]

2.2.2 Securonix

Securonix is a computer tool that helps keep things safe and working well. It looks at a lot of information from different places, like computer networks and systems. Securonix is good at finding and stopping problems quickly. It also helps to make sure everything follows the rules and stays secure. People use Securonix to keep an eye on their computer systems and make sure they are safe from any issues or threats.[7]

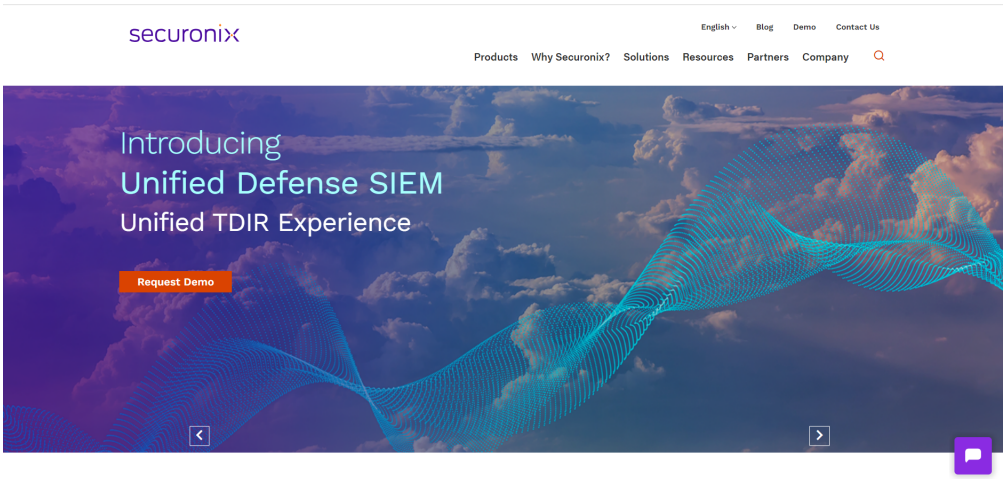

Figure 4: Securonix
[7]

2.2.3 DarkTrace

Artificial intelligence is used by cybersecurity firm Darktrace to quickly identify and address online threats. Their main product, the "Enterprise Immune System," uses machine learning instead of preset rules to detect anomalies and possible threats. [8]

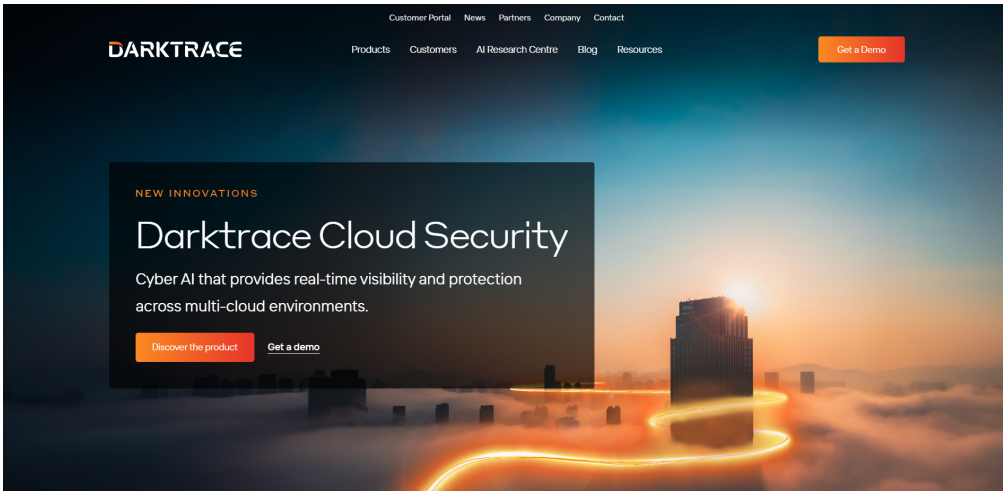

Figure 5: Darktrace
[8]

3 System Description

3.1 Problem Statement

On a daily basis, numerous systems create massive volumes of log data[9]. Analyzing log data is critical for detecting security risks, system failures, and operational concerns. Which means that manual analysis and the security of large organizations is impracticable and time-consuming due to the amount and variety of log data types. In addition to this, The lack of efficient detection tools makes it difficult for IT administrators to manage and maintain the health and performance of complex IT infrastructures. So, Organizations must follow numerous security laws, which frequently need detailed log monitoring and reporting to maintain data integrity and security.

3.2 System Overview

As shown in Figure 6, The system helps in the detection of anomalous actions. It resolved the challenge encountered by the organization about internal threats, which arise when a significant portion of the identical system is computationally expensive and demands substantial quantities of data for efficient training. Particular models may be challenging to interpret. There are SEVEN STAGES in the system:

3.2.1 Data Collection

The first stage collection of logs to construct an all-encompassing data set that accurately reflects the operational environment. This stage entails a systematic approach to log acquisition from diverse origins, encompassing user event logs, server logs, and system logs, all utilized to comprehend user behavior, system operations, and potential security breaches.

3.2.2 Log Parsing

The gathered records undergo the second stage which is a phase of log parsing. This phase facilitates efficient analysis and the unprocessed data is decomposed into structured elements. Nevertheless, the parsing stage is completed, simplifying the interpreting and extracting of significant data.

3.2.3 Feature Extraction

In the third stage, feature extraction, crucial features are extracted from the parsed data to construct a robust feature set. These characteristics contain crucial data that will be utilized in the next stage to train the machine learning algorithms.

3.2.4 Classification

The Machine Learning Algorithms, including Support Vector Machines (SVM), K-means, and Isolation Forests. The algorithms were chosen based on their capacity to analyze intricate data sets and detect patterns in extracted features associated with insider connections. In the data set, the

selected machine learning algorithms train classifiers capable of distinguishing between typical and abnormal patterns. This procedure entails furnishing the algorithm with labeled data to establish precise patterns that can be utilized for classification in the detection phase.

3.2.5 Anomaly Detection

In the fifth phase, the trained classifiers are implemented to detect anomalies in real-time data. The system performs ongoing analysis of incoming records, comparing them to previously learned patterns, and triggers notifications or reactions when anomalies are identified.

3.2.6 Evaluation Matrix

The evaluation matrix stage is mainly used to measure the effectiveness of machine learning models.

3.2.7 Desktop Application

In the seventh and the last stage, the application gives a warning if it detects any abnormal activity. Adopting all these steps will ensure a proactive position in monitoring system performance and security.

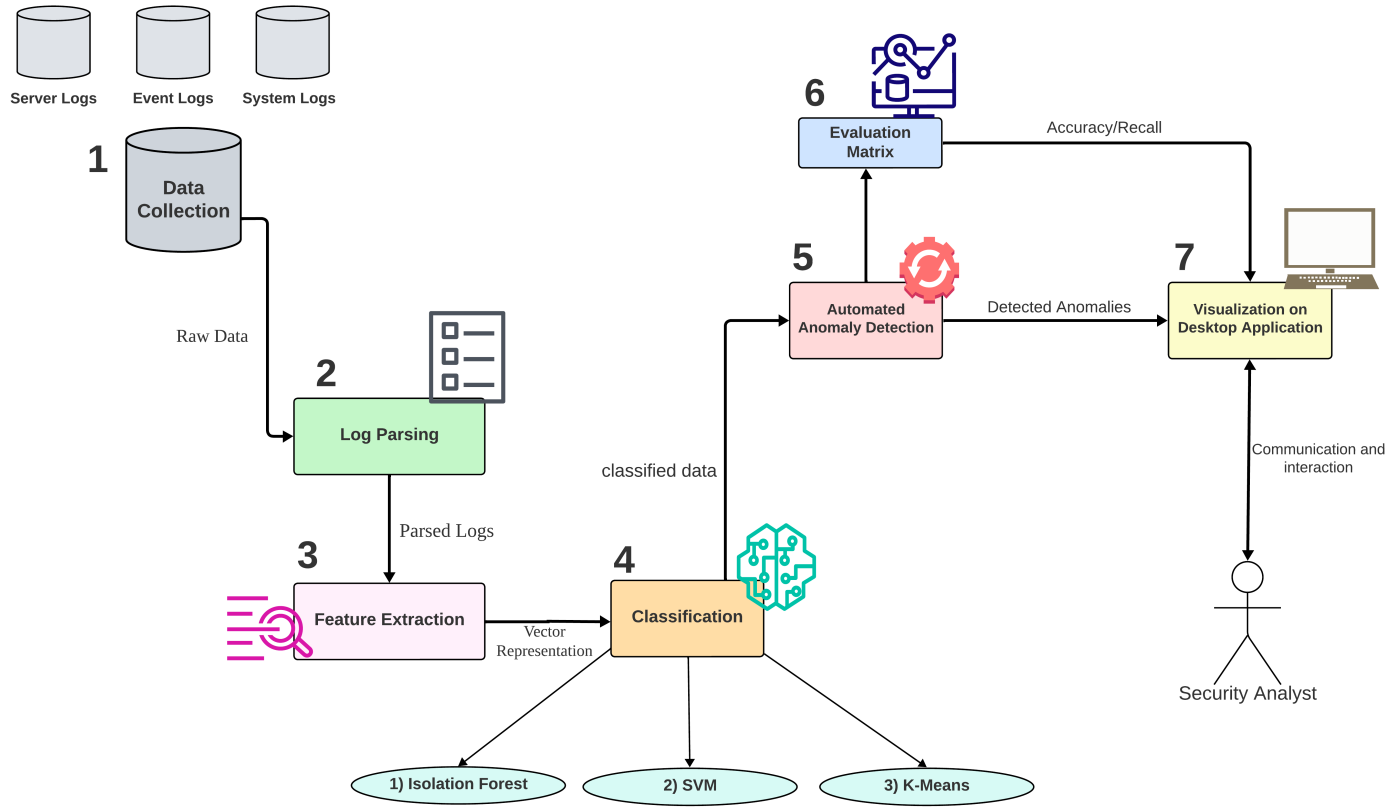

Figure 6: System Overview Diagram

3.3 System Scope

- The system keeps constant monitoring of system's logs.
- Detection of any abnormal behavior happening by an insider.
- Measuring the effectiveness of machine learning models.
- The system sends a warning to stop any threat on the system.
- Securing large corporations by using Anomaly Detection application.

3.4 System Context

Figure 7 represents the context diagram of log anomaly detection application system. The system contains three main entities;Admins ,Security analysts and Classifiers. Firstly, the admin can view the whole system and reports generated by the system and he can modify within the system. Secondly the security analyst can fix vulnerabilities, monitor data and stop or take action. In addition to this, security analyst can view behaviours and get an early warning of any threat.Finally the third entity, which is the classifiers. It's input the data logs and it can classify those data.

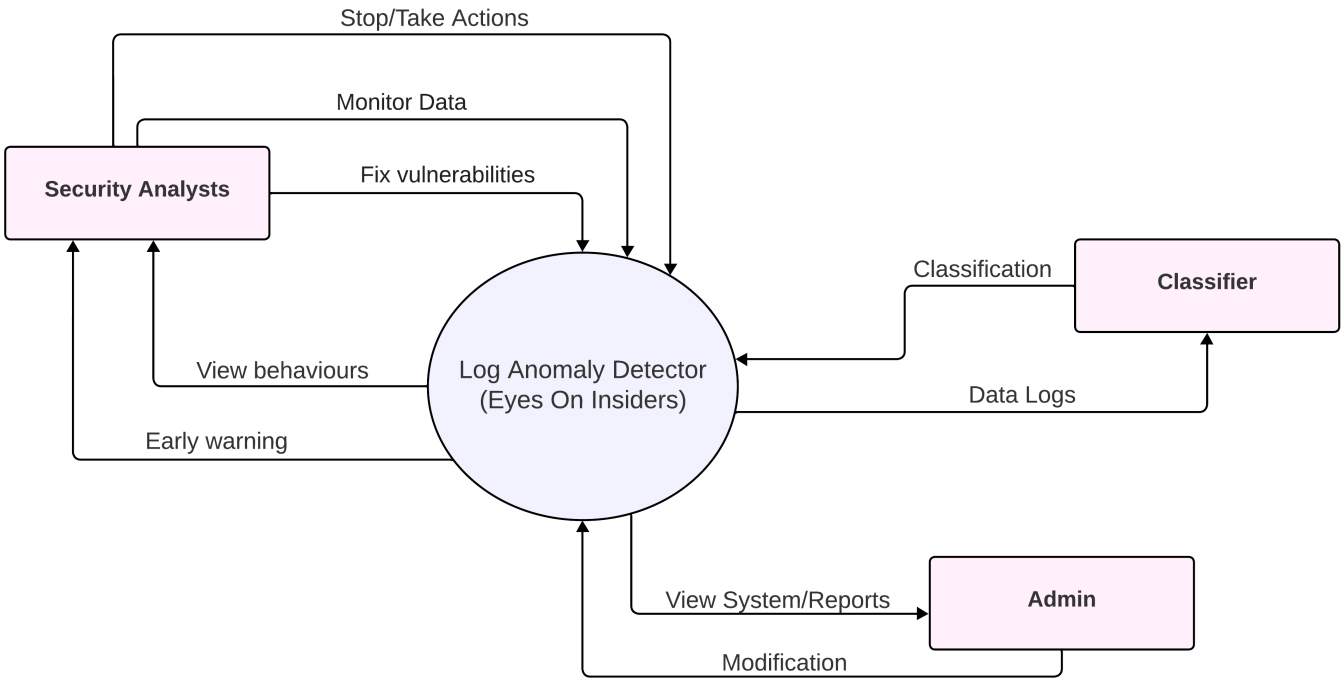

Figure 7: System Context Diagram

3.5 Objectives

The main goals of the system are:

1. Constant Monitoring for log files to detect insiders unexpected behaviors.
2. Determine the difference between normal and abnormal behaviors using machine learning and to secure cooperate's system from the abnormal ones.
3. Get an early warning before system getting attacked.
4. Secure the system from the malicious insiders using the application.

3.6 User Characteristics

- The user must have a strong technical background and expertise in cybersecurity.
- The user should continuously investigate alerts and logs in detail.
- IT administrators should use the log anomaly detection system for monitoring and managing the health and performance of the overall system.

4 Functional Requirements

The use case diagram for this project are shown in figure 8:

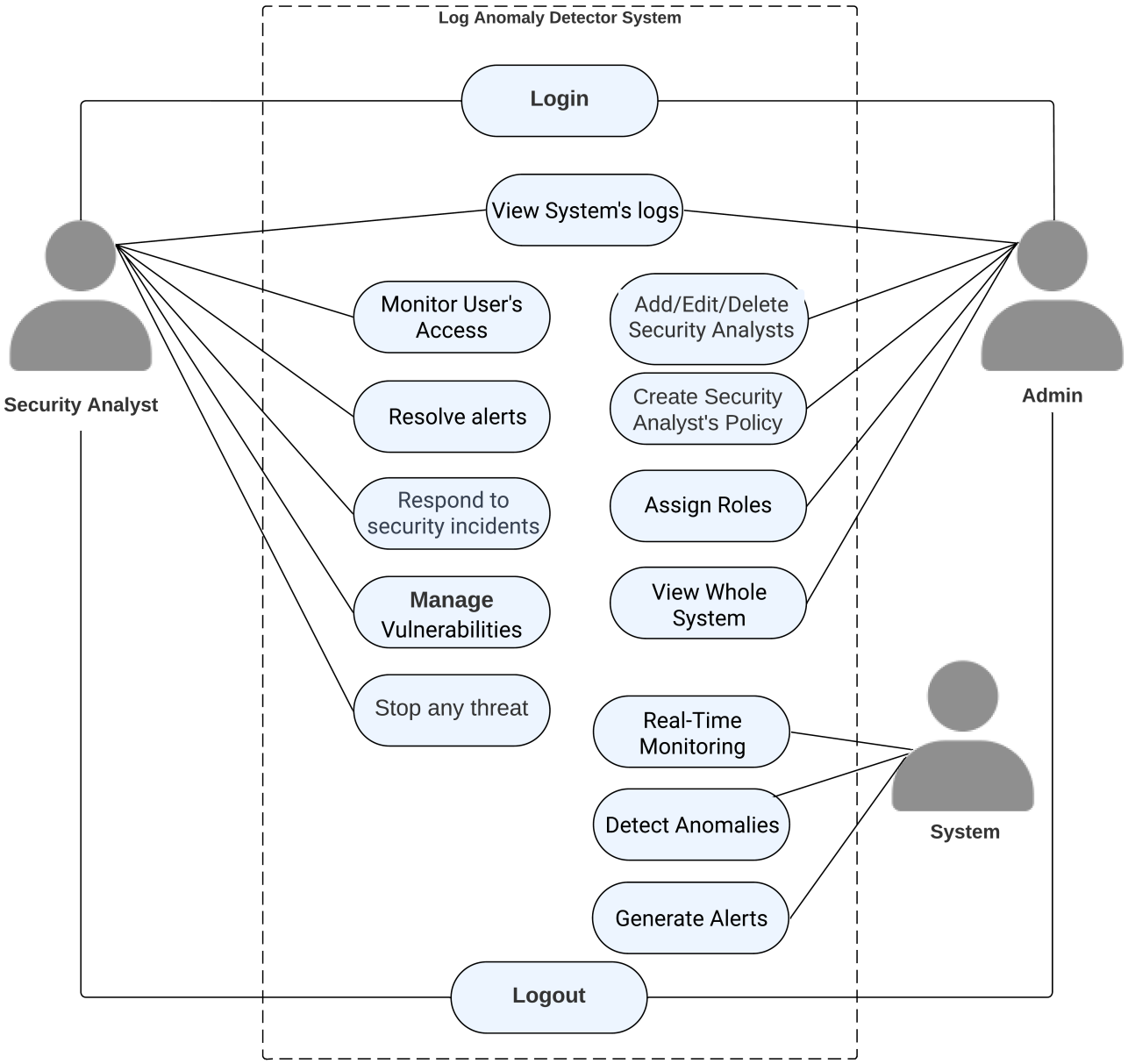

Figure 8: Use Case Diagram

4.1 System Functions

- The System allow real time monitoring.
- The Systen shall detect anomalies.
- The System shall generate alerts.
- The Security Analyst shall parser logs.
- The Security Analyst shall stop any threat.
- The Security Analyst shall manage vulnerabilities.
- The Security Analyst shall respond to security incidents.
- The Security Analyst shall resolve alerts.
- The Security Analyst shall monitor user's access.
- The Security Analyst can view system's logs.
- The Security Analyst can view performance of the detection.
- The Admin can add Security Analysts.
- The Admin can delete Security Analysts.
- The Admin can edit Security Analysts.
- The Admin can view system's logs.
- The Admin can assign roles.
- The Admin can view the whole system.
- The Admin can create policies.

4.2 Detailed Functional Specification

Table 2: Login

Name	**Login**
Code	Fn1
Priority	Extreme
Critical	Cell essential for the user to use all services on the Desktop application
Description	It searches in a database for the entered username and password if it's valid or not
Input	Email and password
Output	Boolean(found or not found)
Pre-condition	User must already have a created account
Post-condition	If found go to the homepage if not say that email or password is incorrect
Dependency	Fn2
Risk	A previous session didn't end

Table 3: Sign Up

Name	**Sign Up**
Code	Fn2
Priority	Extreme
Critical	The data entered by the user must be checked if it has any errors
Description	It checks the data of the user before being sent to the sign up to be added in the database
Input	Email Password First and last name address and mobile number
Output	A string that prints the type of error that occurred
Pre-condition	None
Post-condition	If data has no errors send them to sign up
Dependency	None
Risk	None

Table 4: Parser

Name	**Parser**
Code	Fn3
Priority	High
Critical	None
Description	Simplifies raw logs by cleaning and organizing them for better structure and understanding.
Input	Raw logs
Output	structured logs
Pre-condition	User authentication and authorization
Post-condition	User-defined parser is created and available for use
Dependency	Fn1
Risk	Inaccurate or ineffective parser configurations

Table 5: View System's Logs

Name	**View System's Logs**
Code	Fn4
Priority	High
Critical	None
Description	It allows admin to view system's logs with all its details
Input	Username and Password
Output	Admin's account
Pre-condition	The admin must login
Post-condition	The data will be shown
Dependency	Fn1
Risk	Data Privacy and Unauthorized access

Table 6: Anomaly Detection

Name	**Anomaly Detection**
Code	Fn5
Priority	Extreme
Critical	relies on a clear understanding of normal behavior, dynamic thresholds, real-time processing
Description	Detect anomalies in user behavior using machine learning on log data.
Input	Log files, user activity data
Output	Anomaly detection alerts, reports
Pre-condition	Availability of log data
Post-condition	Improved security through anomaly detection
Dependency	Fn1
Risk	Data privacy concerns

Table 7: Evaluate Performance

Name	**Evaluate Performance**
Code	Fn6
Priority	High
Critical	none
Description	Precision , recall , F1 score , sensitivity and accuracy
Input	Detected anomaly results
Output	Performance of each classifier
Pre-condition	Availability of anomaly detection system
Post-condition	Improved performance of classifier
Dependency	Fn1
Risk	None

5 Design Constraints

5.1 Standards Compliance

The Eyes On Insiders application works on windows operating system.

5.2 Hardware Limitations

The user needs a PC with latest updates.Also, The application's ability to efficiently analyze log data is contingent on the processing power of the CPU. In addition to this, The amount of storage available on the hard drive can impact the application's ability to store and manage log data. Large log data sets require sufficient storage space for both storage and processing.

6 Non-functional Requirements

6.1 Performance Efficiency

Efficiently managing logs in real-time or near real-time is essential, with a focus on minimizing processing delays.

6.2 Scalability

Managing significant volumes of log data becomes imperative as the infrastructure expands, ensuring consistent and dependable performance.

6.3 Reliability

- High Availability: Ensuring continuous operation and minimizing downtime.
- Monitoring and Alerts: Detecting issues in real-time and notifying administrators promptly.

6.4 Security

- Access Control: Restrict and manage user access to log data, ensuring only authorized personnel can view or modify sensitive information.
- Auditing and Logging: Establish and maintain a comprehensive auditing and logging infrastructure within the log anomaly detection system to systematically track and monitor all user activities.

6.5 Usability

The interface should be designed to be user-friendly and intuitive, facilitating administrators and analysts in effortlessly configuring parameters, examining identified anomalies, and implementing necessary measures without requiring specialized expertise.

7 Data Design

In our system, we used two datasets `final_ip_mapped` and `loghub.final_ip_mapped`, comprised of five interconnected tables, meticulously designed to facilitate the anomaly detection process in IP-mapped timestamp data. In the initial phase, the `main_data` table acts as the bedrock, housing the original IP-mapped timestamp data, including key elements such as `@timestamp` and `ip_address`. Subsequently, the `derived_variables` table enhances the dataset by introducing crucial features derived from timestamps, such as `shift_time`, `time_diff`, and `is_weekend`, laying the groundwork for deeper analysis. Moving to Phase 2, the `aggregated_variables` table compiles essential statistics per IP address, offering valuable insights into IP behavior through metrics like `total_count` and `daily_counts`. Phase 3 witnesses the creation of the `merged_data` and `scaled_data` tables, providing a holistic view of IP behavior by integrating aggregated features, clustering results, and scaled subsets for consistency. The journey progresses to Phase 4 with the `kmeans_models` table, storing information about KMeans models to aid in cluster interpretation. Finally, Phase 5 introduces the `performance_metrics` table, capturing quantitative evaluation metrics for different anomaly detection methods, forming a comprehensive and interconnected dataset that serves as the backbone for our graduation project..Secondly , loghub featuring logs from 16 different systems such as distributed systems, supercomputers, and mobile systems, totaling 440 million log messages and 77 GB in size. It serves as a benchmark to assess the accuracy, robustness, and efficiency of existing log parsers.

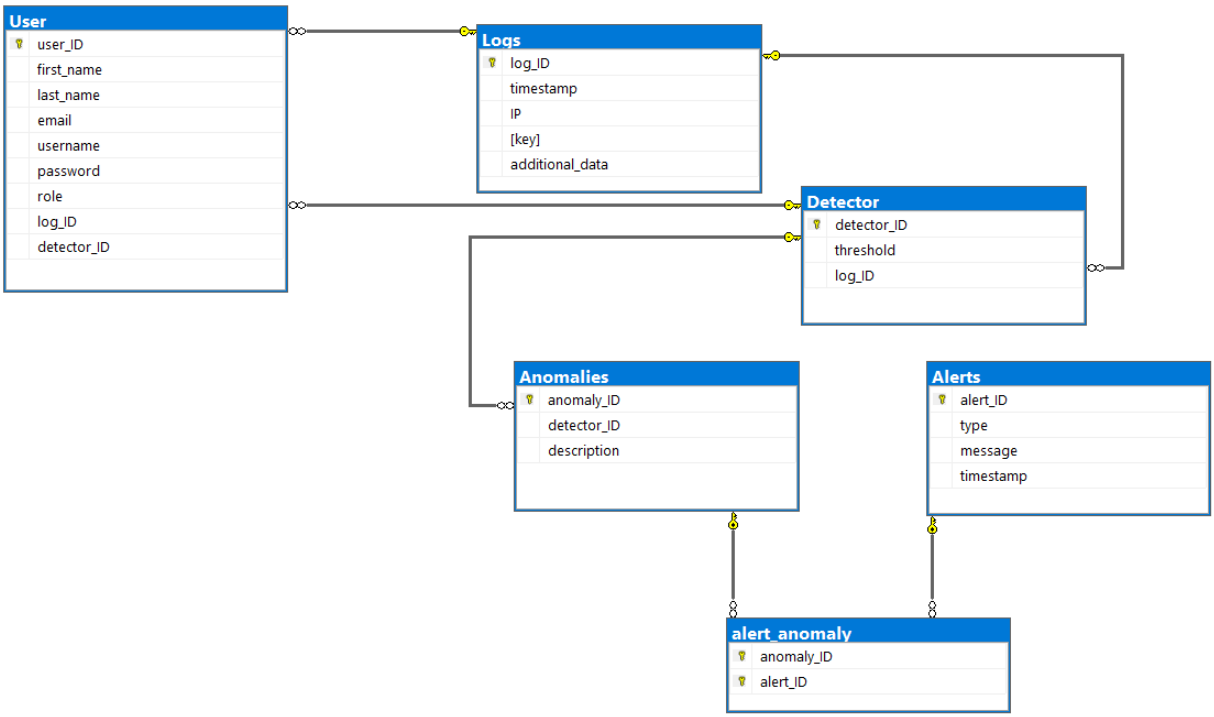

Figure 9: System Database

8 Preliminary Object-Oriented Domain Analysis

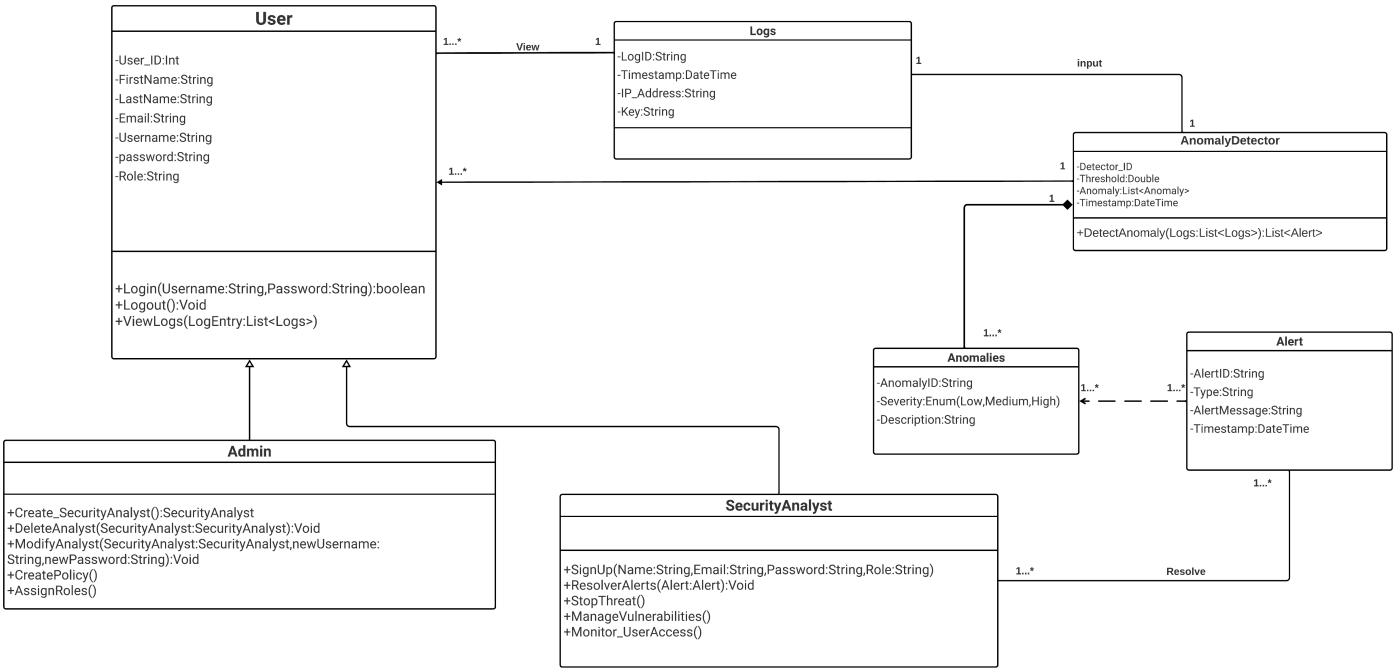

Figure 10: System Class Diagram

9 Operational Scenarios

- **Scenario 1**
  When an ordinary user launches the desktop application, they are granted the ability to effortlessly observe their logon history, which comprises access timestamps and the IP addresses associated with them. The application offers a user-friendly interface that facilitates effective analysis of logs and identification of anomalies. including IP addresses and access timestamps associated with them. The application features a simple user interface that facilitates effective analysis of logs and identification of anomalies.

- **Scenario 2**
  By starting the desktop application, an administrator acquires the ability to effortlessly access the administrative interface. Administrators have the ability to oversee numerous system records, examine the outcomes of anomaly detection, and assess aggregated statistics. Efficient administration is facilitated through the desktop application's seamless navigation and visualization of aberrant system metrics and behaviors.

- **Scenario 3**
  Upon being notified of an anomaly via the desktop application, the user receives immediate access to comprehensive information pertaining to the identified anomaly. By understanding the nature of anomalous behavior, users can be provided with suggested measures to ensure the security of their accounts. A user-friendly experience is ensured when administering security alerts via the application.

- **Scenario 4**
  By employing the desktop application, an administrator examines an anomaly alert and implements appropriate measures, including password resets and IP address barring, in response. Admins are granted the authority to respond to anomalies, implement security measures, and communicate with affected users via the dedicated desktop interface. This enables the administration to streamline tasks in order to provide efficient user support and response. After administrative intervention, the desktop application ensures an exhaustive audit trail by logging the actions performed in a secure local database. The database retains pertinent information for future reference, including the timestamp of administrative responses, the types of actions performed, and any supplementary comments. This information aids in the continuous monitoring of compliance and security.

10 Project Plan

The figures 11 , 12 below shows the timeline of this project from the proposal to SDD .

Task	Start date	End date	Duration in days	Role
Information collection and researches	9/20/2023	10/20/2023	30	All Team Members
Survey and proposal preparation	10/20/2023	11/5/2023	16	All Team Members
Preprocessing stage	11/6/2023	11/12/2023	6	All Team Members
Proposal presentation 10%	11/13/2023	11/16/2023	2	All Team Members
Documentation of SRS	12/16/2023	1/10/2024	31	All Team Members
Increasing Dataset	11/16/2023	12/31/2023	45	Jana and Hoda
Testing Classifiers	11/20/2023	11/29/2023	9	Yasmin and Mohand
Outline Log Parsing	1/1/2024	1/10/2024	10	All Team Members
Increasing Accuracy of Detection	1/3/2024	1/12/2024	9	All Team Members
PowerPoint of SRS	1/10/2024	1/15/2024	5	All Team Members
Writing Survey Paper	1/4/2024	1/15/2024	11	All Team Members
Developing the Desktop Application	1/12/2024	3/12/2024	60	All Team Members
Documentation of SDD	1/20/2024	3/10/2024	50	All Team Members

Figure 11: SRS Timeline

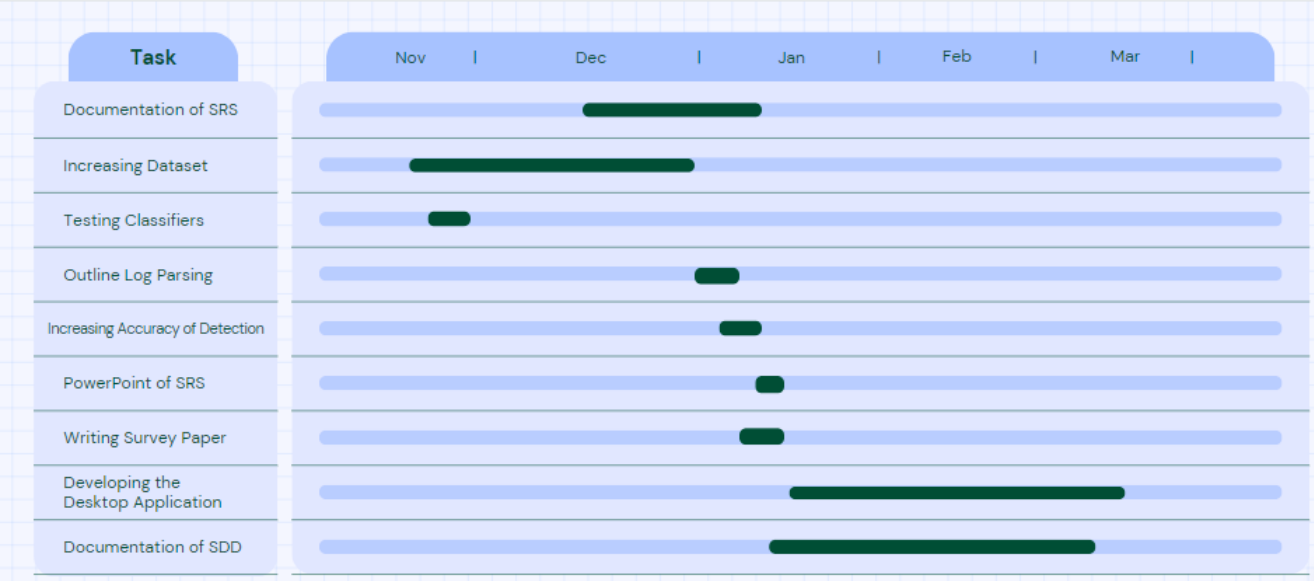

Figure 12: SRS Timeline Gantt Chart

11 Appendices

11.1 Definitions, Acronyms, Abbreviations

Abbreviations	Description
SVM	Supportive vector machine
K-means	the number of clusters (groups) that we want to form.
Isolated Forest	a technique for identifying outliers in data
Log parsing	the systematic analysis and extraction of specific information from log files
Precariousness	the state of being dangerously likely to fall or collapse
Final IP mapped data	This dataset is crucial for anomaly detection in IP-mapped timestamp data
Loghub	Large collection of logs

11.2 Supportive Documents

- Dataset:

  1- Final IP mapped data : dataset consists of five interconnected tables meticulously designed for anomaly detection in IP-mapped timestamp data, encompassing original data, derived features, aggregated statistics, merged and scaled subsets, three classifiers models, and performance metrics.
  2-Loghub [10] : A large collection of logs from 16 different systems spanning distributed systems, supercomputers, operating systems, mobile systems, server applications, and standalone software.

- Survey

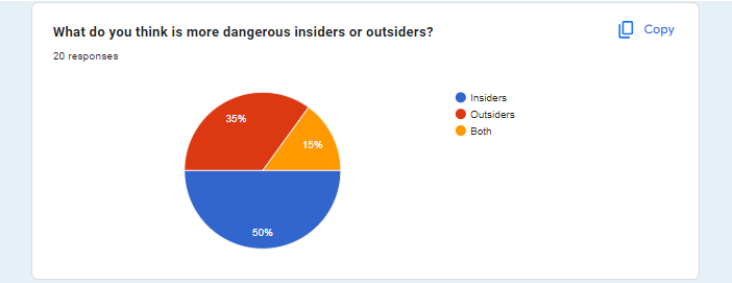

Figure 13: Question 1

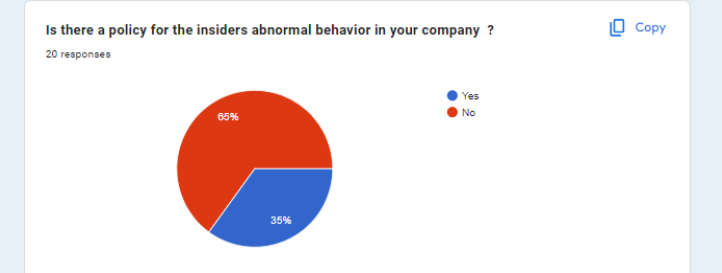

Figure 14: Question 2

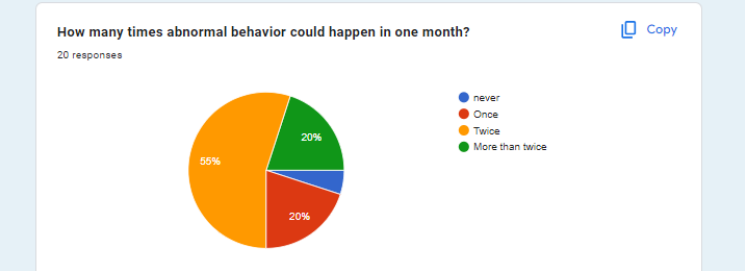

Figure 15: Question 3

References

[1] Aram Kim, Junhyoung Oh, Jinho Ryu, et al. "A Review of Insider Threat Detection Approaches With IoT Perspective". In: *IEEE Access* 8 (2020), pp. 78847–78867. DOI: 10.1109/ACCESS.2020.2990195.

[2] Shangbin Han, Qianhong Wu, Han Zhang, et al. "Log-based anomaly detection with robust feature extraction and online learning". In: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 2300–2311.

[3] Jieming Zhu, Shilin He, Jinyang Liu, et al. "Tools and benchmarks for automated log parsing". In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE. 2019, pp. 121–130.

[4] Guansong Pang, Chunhua Shen, Longbing Cao, et al. "Deep learning for anomaly detection: A review". In: *ACM computing surveys (CSUR)* 54.2 (2021), pp. 1–38.

[5] Mohammad S Islam, William Pourmajidi, Lei Zhang, et al. "Anomaly detection in a large-scale cloud platform". In: *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE. 2021, pp. 150–159.

[6] *Splunk | The Key to Enterprise Resilience — splunk.com*. https://www.splunk.com/. [Accessed 10-11-2023].

[7] *Securonix: Security Analytics at Cloud Scale. — securonix.com*. https://www.securonix.com/. [Accessed 10-11-2023].

[8] *Darktrace | Cyber security that learns you — darktrace.com*. https://darktrace.com/. [Accessed 10-11-2023].

[9] Rakesh Bahadur Yadav, P Santosh Kumar, and Sunita Vikrant Dhavale. “A survey on log anomaly detection using deep learning”. In: *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE. 2020, pp. 1215–1220.

[10] Jieming Zhu, Shilin He, Pinjia He, et al. “Loghub: A large collection of system log datasets for ai-driven log analytics”. In: *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE. 2023, pp. 355–366.