



ATTENTION-SEEKING MODE WITH A SCORE SYSTEM USING MACHINE LEARNING

Project Proposal Report

DEPARTMENT OF INFORMATION
TECHNOLIGY

Pubudu Priyanga Liyanage
Cyber Security

Risk Scoring System for Data Loss Prevention

RP24_25J_003

Project Proposal Report

Pubudu Priyanga Liyanage

B.Sc. (Hons) Degree in Information Technology specialized in
Cyber Security


Department of Information Technology

Sri Lanka Institute of Information Technology

June 2024

Declaration

I declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Group member name	Student ID	Signature
Liyanage P.P.	IT21184758	

The above candidate is carrying out research for the undergraduate Dissertation under supervision of the undersigned.

.....

Signature of the supervisor

(Mr. Amila Senerathna)

.....

Date

.....

Signature of Co-supervisor

(Ms. Suranjini Silva)

.....

Date

Acknowledgement

I extend my sincere gratitude to my supervisor, Mr. Amila Senerathna, and co-supervisor, Ms. Suranjini Silva, for their invaluable guidance and support throughout this research study. I'm thankful to industry experts for sharing their expertise which helped me get a certain domain knowledge. Special thanks to my team members for their contributions, and to those who aided me willingly. Lastly, my heartfelt appreciation to my family for their constant love, assistance, and encouragement.

Abstract

The volume and complexity of security alerts generated by conventional Data Loss Prevention (DLP) systems often lead to alert fatigue, causing delays or oversights in addressing critical incidents. This research aims to mitigate this issue by developing an intelligent system that prioritizes alerts using a risk score mechanism powered by machine learning. The proposed system will analyze historical data and real-time incidents to assign scores based on impact and relevance, enabling security teams to focus on high-priority threats efficiently. The outcome will be an adaptive, efficient, and user-friendly alert prioritization module that integrates seamlessly into existing DLP tools, improving incident response times and reducing operational fatigue.

Keywords : Alert Fatigue, Machine Learning, Data Loss Prevention, Risk Score, Incident Prioritization, Cybersecurity.

Table of Contents

Declaration.....	2
Acknowledgement.....	3
Abstract	4
Table of Contents	5
List of Abbreviations.....	7
1. Introduction.....	7
2. Background and literature survey	8
3. Research Gap.....	10
4. Research Problem.....	10
5. Objectives	11
5.1. Main Objective.....	11
5.2. Sub Objectives	11
6. Methodology	12
6.1. System Architecture	12
6.2. Software solution.....	14
6.2.1. Requirements gathering and Analysis	15
6.2.2. Feasibility study	16
6.2.3. Dataset Preparation.....	16
6.2.4. Implementation	16
6.2.5. Testing.....	16
6.2.6. Deployment.....	17
6.3. Tools and Techniques	17
7. Project Requirements	18
7.1. Functional Requirements	18
7.2. Non-Functional Requirements	18
7.3. System Requirements	18

7.4. User Requirements.....	19
8. References.....	21

List of Abbreviations

Abbreviation	Description
DLP	Data Loss Prevention
SDLC	Software Development Life Cycle
PII	Personally Identifiable Information
UAT	User Acceptance Testing

1. Introduction

The digital era has ushered in an unprecedented increase in cyber threats, necessitating robust Data Loss Prevention (DLP) systems to protect sensitive data. However, the effectiveness of these systems is often undermined by the overwhelming number of alerts they generate, leading to what is termed as “alert fatigue” among security teams. This fatigue can result in delayed responses to critical incidents or the overlooking of high-risk alerts, compromising an organization’s security posture.[4]

This research proposes a novel approach to address this challenge by incorporating an attention-seeking mode with a machine learning-based score system within DLP frameworks. By prioritizing alerts based on their calculated risk score, the system aims to enhance incident response efficiency and reduce operational fatigue. The solution leverages historical data and incident characteristics to dynamically prioritize security alerts, ensuring timely and effective responses to high-risk incidents. [6]

2. Background and literature survey

The exponential growth of digital transformation has reshaped how organizations manage their security postures, emphasizing the need for effective Data Loss Prevention (DLP) systems.

However, traditional DLP tools struggle with challenges such as high volumes of alerts and limited prioritization mechanisms, often leading to operational inefficiencies and alert fatigue. Security teams are inundated with numerous low-priority alerts, making it difficult to identify and respond to critical incidents promptly.[5]

Alert Fatigue and its Impact: Alert fatigue refers to the desensitization of security teams to alerts due to their sheer volume and perceived lack of relevance. Studies suggest that over 75% of security alerts are ignored by teams, potentially allowing high-risk threats to go unnoticed. This underscores the need for systems that can intelligently filter and prioritize alerts based on their severity and impact.[6]

Machine Learning in Cybersecurity: The advent of machine learning has introduced transformative capabilities in cybersecurity, particularly in anomaly detection and threat prioritization. Supervised learning algorithms, such as decision trees and support vector machines (SVM), have been utilized to classify incidents based on historical patterns. Unsupervised learning methods, including clustering and dimensionality reduction techniques, enable the detection of anomalies in complex datasets, making them invaluable for dynamic environments.[7]

Advancements in Risk Scoring: Risk scoring systems have evolved significantly, leveraging both historical data and real-time inputs to assign impact-based scores to alerts. For instance, predictive models trained on labeled datasets can identify recurring threat patterns and estimate their potential consequences. By incorporating factors such as asset criticality, incident frequency, and contextual data, these models can generate actionable insights for security teams.[8]

Dynamic Prioritization: Traditional rule-based prioritization methods are often static and fail to adapt to evolving threats. Machine learning enables dynamic prioritization by continuously

learning from new data and refining its scoring mechanisms. This ensures that high-risk incidents are addressed promptly, reducing the likelihood of breaches.

Integration Challenges: Despite their potential, integrating machine learning-based solutions into existing DLP systems presents challenges, including computational overhead, compatibility issues, and the need for large labeled datasets. Addressing these challenges requires the development of lightweight models that balance accuracy and performance.

Case Studies and Applications: Various case studies highlight the effectiveness of machine learning in alert prioritization. For example, a 2023 study on financial institutions demonstrated a 60% reduction in response times by implementing machine learning-based scoring systems. Another study in healthcare cybersecurity showed improved incident detection rates through real-time prioritization mechanisms.[6]

Emerging Trends: The integration of natural language processing (NLP) and deep learning techniques is emerging as a promising trend. NLP can analyze textual data in alerts to extract contextual information, while deep learning models such as convolutional neural networks (CNNs) offer enhanced capabilities for analyzing complex patterns in large datasets.

This research builds on these advancements by developing a machine learning-driven attention-seeking mode that prioritizes alerts based on risk scores. By addressing the limitations of existing systems, the proposed solution aims to enhance the efficiency of DLP frameworks and improve organizational resilience against cyber threats.[8]

3. Research Gap

As discussed in the above literature review, a notable gap can be seen in the existing systems. Below mentioned are the research gaps found.

Application References	Real-time Risk Scoring	Alert Fatigue Reduction	Historical Data Analysis	Critical Incident Prioritization
Reference [1]				
Reference [2]				
Reference [3]				
DATASHIELDX				

4. Research Problem

Conventional Data Loss Prevention (DLP) systems are essential for monitoring and protecting sensitive information; however, their inherent design often results in an overwhelming volume of alerts. This deluge of notifications not only overburdens security teams but also creates a phenomenon known as alert fatigue, where the sheer quantity of alerts leads to delays in response times and increases the likelihood of critical threats being overlooked. This situation is exacerbated in dynamic environments where real-time responses are critical to mitigating risks effectively.

To address these challenges, there is a pressing need for an intelligent system capable of not only analyzing alerts in real time but also assigning accurate risk scores to each alert. By leveraging advanced analytical capabilities, such a system can prioritize alerts based on their potential impact and relevance. This prioritization ensures that security teams can focus their attention and resources on high-priority threats, enhancing their efficiency and reducing the risk of missing crucial incidents.

This research aims to tackle these pressing issues by developing a machine learning-based alert prioritization module. This module will utilize sophisticated algorithms to process and evaluate alert data, assigning dynamic risk scores based on contextual and historical information. Through this approach, the system will provide actionable insights, streamlining decision-making processes for security teams and ultimately improving organizational resilience against cyber threats.

5. Objectives

5.1. Main Objective

To develop a robust machine learning-based system capable of prioritizing security alerts by assigning risk scores, enabling security teams to focus their efforts on addressing the most critical threats effectively.

5.2. Sub Objectives

1. **Detection and Classification of Sensitive Information:**

Design and implement a machine learning-based system that can accurately detect and classify sensitive information contained within security alerts. This involves training models on labeled datasets to recognize patterns, categories, and context associated with sensitive data, ensuring comprehensive coverage across various alert types and sources.

2. **Risk Scoring of Security Alerts:**

Develop a methodology for assigning dynamic risk scores to security alerts. This scoring system will leverage severity levels, potential business impact, and historical incident data to quantify the urgency and importance of each alert. The scoring algorithm will integrate real-time analysis and contextual factors to provide a nuanced understanding of risk.

3. **System Effectiveness Evaluation:**

Conduct rigorous testing and validation of the system to assess its effectiveness in mitigating alert fatigue and improving response prioritization. This includes:

- Measuring the reduction in the volume of alerts requiring immediate attention.
- Analyzing improvements in response times for high-risk incidents.

- Gathering feedback from security teams on usability and performance.
- Comparing results with traditional DLP systems to evaluate the impact of machine learning integration.

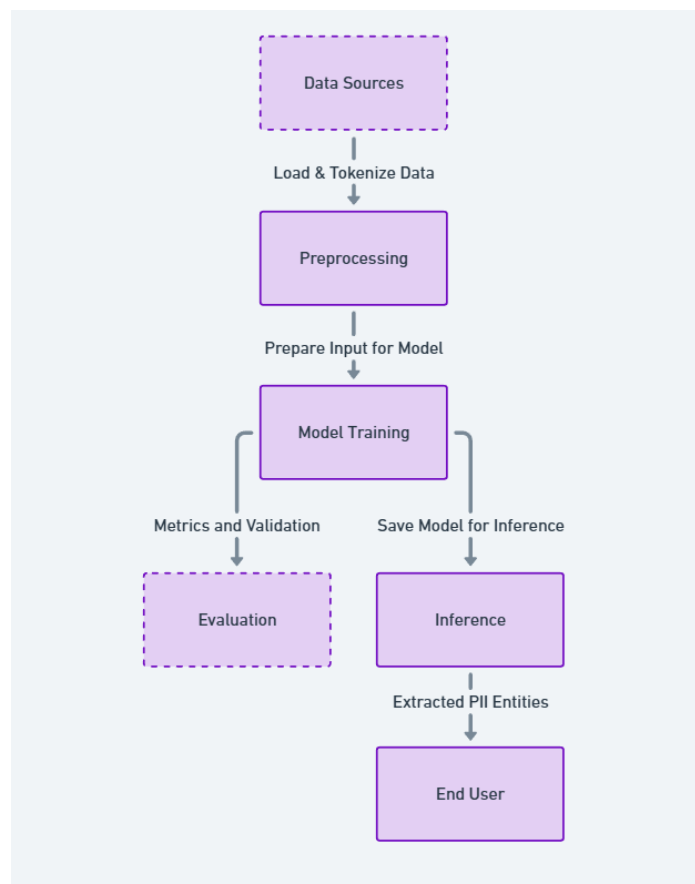
Through these objectives, the project aims to create an intelligent solution that streamlines alert management processes, optimizes resource allocation, and enhances the overall cybersecurity posture of an organization.

6. Methodology

6.1. System Architecture

The projects' main goal is to provide an overall solution based on data loss prevention which facilitates in mitigating sensitive data leakage or data leakage attacks in the organization.

The way which the risk scoring component works is illustrated below.



The system architecture for the project revolves around building a machine learning-based pipeline designed to address alert fatigue in Data Loss Prevention (DLP) systems by leveraging Named Entity Recognition (NER). The architecture integrates various interconnected components, each serving a specific purpose in the overall workflow.

The foundation of the system begins with the data collection process, which involves gathering historical incident and alert data. This dataset includes both labeled and unlabeled examples, ensuring a comprehensive representation of possible scenarios in a real-world DLP system. The labeled data contains specific tags corresponding to sensitive information entities such as email addresses, phone numbers, and other identifiers, while the unlabeled data provides a testing ground for model evaluation.

Once the data is gathered, preprocessing is carried out to prepare it for the machine learning model. A tokenizer is utilized to convert the raw text into tokens, the basic units of language processing. This tokenizer ensures that the data adheres to the input format required by the transformer-based model. Additionally, the system incorporates a mapping mechanism to assign numerical identifiers to each label, enabling efficient computation during model training. This mapping also allows for easy interpretation of model outputs by converting numerical predictions back into human-readable labels.

The next stage of the pipeline involves training the machine learning model. A pre-trained transformer model, specifically designed for token classification, is fine-tuned on the prepared dataset. Fine-tuning enables the model to learn domain-specific patterns and improve its accuracy in identifying sensitive information. To enhance the training process, various optimization strategies such as learning rate scheduling, gradient accumulation, and warm-up ratios are implemented. The training process includes dividing the dataset into training and validation subsets to evaluate the model's performance and avoid overfitting.

Evaluation metrics play a crucial role in assessing the effectiveness of the trained model. Metrics such as precision, recall, and F1-score are computed to measure the model's ability to accurately detect and classify sensitive entities. These metrics ensure that the model balances its performance between minimizing false positives and maximizing true positives, which is critical in a cybersecurity context.

After training, the system enters the inference stage. The trained model is deployed to classify sensitive information in new, unseen data. During this stage, the input text is tokenized and processed similarly to the training phase. The model then generates predictions, assigning confidence scores to each identified entity. Post-processing ensures that the output is coherent and actionable, enabling security teams to focus on high-priority alerts.

The architecture also incorporates iterative optimization. Feedback from the evaluation and inference stages is used to refine the model further. Adjustments to the training data, hyperparameters, or feature engineering strategies are made based on observed performance, ensuring continuous improvement.

In essence, this architecture forms a robust and intelligent system capable of reducing alert fatigue by prioritizing high-risk alerts and providing actionable insights. The integration of machine learning with domain-specific optimizations ensures that the system meets the demands of modern cybersecurity environments, ultimately enhancing the efficiency of security teams.

6.2. Software solution

Development will be done iteratively; the DLP tool's Agile Software Development Lifecycle using SCRUM methodology will be followed for the development of the risk scoring component. This approach gives much emphasis to continuous integration and collaboration between developers and stakeholders for the efficiency and adaptability of the component.

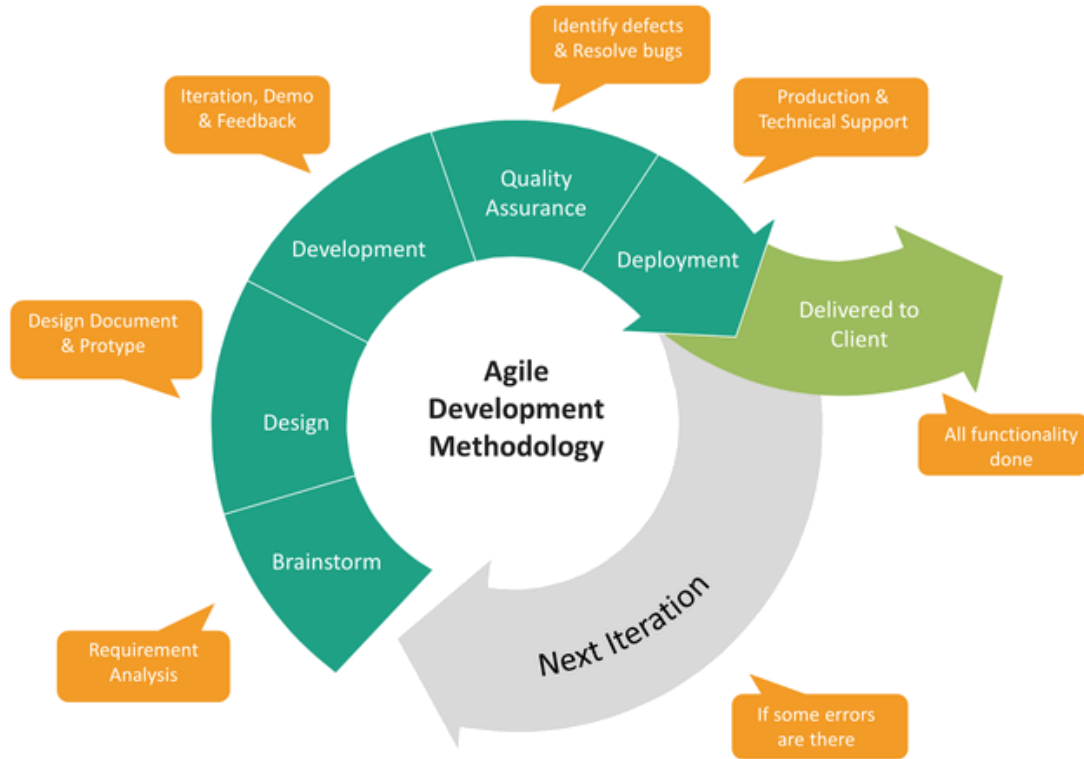


Figure 1-Agile methodology

6.2.1. Requirements gathering and Analysis

This stage focuses on ensuring that the score-based prioritization component for the DLP tool aligns with organizational needs and cybersecurity objectives. It begins with identifying key stakeholders, such as security analysts and developers, to understand their pain points and gather input for system design. The functional requirements include the capability to process alerts, assign risk scores based on severity, and prioritize the most critical alerts for immediate attention. Additionally, the system must provide real-time notifications and generate actionable insights. Non-functional requirements emphasize high accuracy, real-time performance, scalability for large-scale environments, and adherence to data privacy standards.

Datasets will be collected, including historical alert data with labeled features for training the machine learning model. During this phase, features like source, destination, anomaly type, and previous incident impact are analyzed to identify factors contributing to alert risk. This structured requirement analysis ensures the system is robust and capable of reducing alert fatigue while meeting industry best practices.

6.2.2. Feasibility study

A feasibility study evaluates the technical and economic viability of the proposed solution. This includes assessing computational resources required to process high volumes of alerts in real-time and the cost of implementing machine learning models and infrastructure. The development team will explore whether the solution can integrate seamlessly with existing DLP systems and whether it meets budgetary constraints.

6.2.3. Dataset Preparation

Historical alert data will be curated to include features like alert type, frequency, and severity. Additional datasets with incident classifications and risk attributes will also be created. This data is essential for training machine learning models to assign risk scores and prioritize alerts. Data augmentation techniques may be applied to ensure the dataset is diverse and representative of real-world scenarios.

6.2.4. Implementation

The implementation phase includes several critical steps:

- **Model Training:** Develop and train a machine learning model capable of evaluating alert features and assigning risk scores.
- **Risk Scoring Mechanism:** Implement a scoring system based on the model's predictions to rank alerts by severity and potential impact.
- **Real-Time Integration:** Integrate the scoring system into the DLP infrastructure to evaluate incoming alerts and prioritize them for response.
- **Alert Feedback Mechanism:** Design a feedback loop to allow security teams to provide input, refining the scoring algorithm over time.

6.2.5. Testing

To ensure functionality, the following tests will be conducted:

- **Unit Testing:** Validate individual components of the scoring and prioritization system.
- **Integration Testing:** Ensure smooth integration with existing DLP systems.
- **System Testing:** Test the entire solution, including data ingestion, scoring, and alert generation.
- **User Acceptance Testing (UAT):** Security analysts and incident response teams will verify that the system meets operational needs and improves efficiency.

6.2.6. Deployment

The system will be deployed in a live environment, leveraging cloud infrastructure for scalability and reliability. The machine learning models and scoring algorithms will operate on a server capable of handling high-volume alert streams, ensuring real-time performance.

6.3. Tools and Techniques

Tools

- **Python:** For implementing machine learning algorithms and integrating the system.
- **Anaconda:** To manage environments and dependencies during development.
- **Jupyter Notebooks:** For prototyping and testing machine learning models.
- **Flask/Django:** For developing the web-based interface to visualize prioritized alerts.
- **AWS/GCP/Heroku:** For deploying the system in a scalable cloud infrastructure.
- **GitHub:** For version control and collaborative development.
- **Microsoft Planner:** To track project tasks and milestones.

Techniques

- **Machine Learning Models:** Train models to predict alert severity and assign risk scores.
- **Feature Engineering:** Identify key alert attributes that influence risk scores.
- **Feedback Loops:** Use analyst input to refine model predictions over time.
- **Data Augmentation:** Enhance datasets with synthetic or derived features for robust training.
- **Testing and Validation:** Conduct thorough testing to ensure quality and reliability.

7. Project Requirements

7.1. Functional Requirements

- The system shall analyze incoming alerts in real-time and assign risk scores based on predefined features.
- It shall prioritize critical alerts, ensuring immediate action by security teams.
- The system shall generate detailed notifications, including alert type, severity, and confidence levels.
- A user-friendly interface will display prioritized alerts and provide interactive visualizations.
- The system shall log all alert-related actions and generate periodic performance reports.

7.2. Non-Functional Requirements

- The system must scale to handle high volumes of alerts without performance degradation.
- It should maintain an accuracy rate above 95% in predicting alert severity and risk.
- Data privacy must be ensured, with secure handling of all alert data.
- The interface should be intuitive, enabling security analysts to quickly interpret alert priorities.
- The system must be modular, allowing updates and new features to be added with minimal disruption.

7.3. System Requirements

The system requirements outline the software, hardware, and infrastructure necessary to support the implementation of the machine learning-based alert prioritization tool. The following components are essential for ensuring efficient operation, scalability, and integration with existing Data Loss Prevention (DLP) systems:

1. Software Requirements:

- **Programming Environment:** Python will serve as the primary programming language due to its rich libraries and frameworks for machine learning and data processing. Anaconda will manage dependencies and virtual environments to maintain a streamlined and conflict-free development workflow.
- **Development Tools:** A robust Integrated Development Environment (IDE) such as PyCharm or Visual Studio Code is required to facilitate efficient coding, debugging, and testing. These tools offer features such as version control, syntax

highlighting, and code suggestions, which enhance productivity during the development phase.

- **Web Frameworks:** Flask or Django may be used for creating a user-friendly web-based interface. These frameworks enable rapid prototyping, seamless integration with backend logic, and visualization of alert prioritization results.
- **Cloud Platforms:** Hosting and deployment will require cloud services such as AWS, Google Cloud Platform (GCP), or Heroku. These platforms provide scalable and reliable infrastructure capable of handling large volumes of alert data.

2. **Hardware Requirements:**

- **Processing Power:** The system will require a high-performance server with a multi-core CPU and GPU capabilities to support training and inference of machine learning models. The use of GPUs significantly accelerates the training of complex models by parallelizing computations.
- **Storage:** Adequate storage is essential for datasets, logs, and system configurations. The system will utilize SSDs for faster read/write speeds, especially when processing large datasets during real-time operations.
- **Networking:** High-speed and reliable internet connectivity is necessary to support cloud interactions, data streaming, and real-time processing of alerts.

3. **Integration Requirements:**

- The system must integrate seamlessly with existing DLP tools, enabling it to process incoming alerts without disrupting workflows.
- APIs will be required to facilitate communication between the alert prioritization module and the DLP infrastructure. This includes retrieving alerts, sending prioritized results, and triggering notifications.

4. **Scalability:**

- The architecture must support horizontal scaling to accommodate increasing data loads and organizational growth. This ensures that the system can handle higher alert volumes without performance degradation.

5. **Security:**

- Data security is a priority, and the system must implement encryption for data in transit and at rest.
- Role-based access control (RBAC) must be enforced to restrict unauthorized access to sensitive information and system configurations.

7.4. User Requirements

The system is designed to address the needs of different user groups, including security analysts, incident response teams, system administrators, and compliance officers. Each group has distinct requirements that the system must fulfill to ensure usability, efficiency, and compliance with industry standards:

1. Security Analysts:

- **Real-Time Alert Prioritization:** Analysts need the system to deliver a ranked list of alerts based on risk scores, enabling them to focus on the most critical threats first.
- **Detailed Alert Insights:** Each prioritized alert should include details such as type, severity, source, destination, and confidence levels to provide actionable context.
- **Interactive Dashboard:** A user-friendly interface must display real-time alerts, system status, and historical trends. Analysts should be able to filter, sort, and analyze alerts using the dashboard.
- **Configurable Notifications:** Alerts should be sent via email, SMS, or integrated messaging systems, allowing analysts to stay informed even when away from their workstations.

2. Incident Response Teams:

- **Automated Threat Mitigation:** The system must block high-risk threats automatically, such as quarantining files or restricting network access, minimizing the need for manual intervention.
- **Incident Context:** Response teams require access to comprehensive logs that detail the origin, progression, and resolution of each detected threat.
- **Customizable Workflows:** The tool must support integration with existing incident response workflows, enabling teams to trigger predefined actions for high-priority alerts.

3. System Administrators:

- **Access Control:** Administrators must be able to manage user roles and permissions to ensure secure access to the system.
- **System Configuration:** The system should offer a configurable interface to adjust detection parameters, alert thresholds, and notification preferences.
- **Maintenance and Updates:** Administrators need tools for system diagnostics, updates, and patches to ensure uninterrupted functionality. Regular updates to the machine learning models should also be supported to adapt to emerging threats.

4. Compliance Officers:

- **Audit Logs:** The system must maintain tamper-proof logs of all detection events, user actions, and system activities. These logs should be exportable for compliance audits and reporting.
- **Comprehensive Reports:** The system should generate detailed reports summarizing detection trends, system performance, and compliance metrics. These reports must meet regulatory standards and be suitable for submission to external auditors.
- **Data Privacy Assurance:** Compliance officers require guarantees that the system adheres to data protection regulations, such as encrypting sensitive data and minimizing the storage of unnecessary personal information.

By addressing these diverse user requirements, the system ensures that all stakeholders can effectively utilize the tool to enhance organizational cybersecurity.

8. References

- [1] O. Keskin, N. Gannon, B. Lopez and U. Tatar, "Scoring Cyber Vulnerabilities based on Their Impact on Organizational Goals," 2021 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2021, pp. 1-6, doi: 10.1109/SIEDS52267.2021.9483741.
- [2] E. Seker and W. Meng, "XVRS: Extended Vulnerability Risk Scoring based on Threat Intelligence," 2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom), Kyoto, Japan, 2023, pp. 516-523, doi: 10.1109/MetaCom57706.2023.00094.
- [3] M. Keramati, "New Vulnerability Scoring System for dynamic security evaluation," 2016 8th International Symposium on Telecommunications (IST), Tehran, Iran, 2016, pp. 746-751, doi: 10.1109/ISTEL.2016.7881922.
- [4] S. Ndichu, T. Ban, T. Takahashi and D. Inoue, "Critical-Threat-Alert Detection using Online Machine Learning," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 3007-3014, doi: 10.1109/BigData55660.2022.10021115.
- [5] K. Shaukat, S. Luo, S. Chen and D. Liu, "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective," 2020 International Conference on Cyber Warfare and Security (ICWS), Islamabad, Pakistan, 2020, pp. 1-6, doi: 10.1109/ICWS48432.2020.9292388.
- [6] S. Ndichu, T. Ban, T. Takahashi and D. Inoue, "A Machine Learning Approach to Detection of Critical Alerts from Imbalanced Multi-Appliance Threat Alert Logs," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 2119-2127, doi: 10.1109/BigData52589.2021.9671956.
- [7] R. Ch, B. Naresh, P. L. Prasanna, N. Chander, E. A. Goud and P. R. Prasad, "Exploring Machine Learning Algorithms for Robust Cyber Threat Detection and Classification: A Comprehensive Evaluation," 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), Gurugram, India, 2024, pp. 01-05, doi: 10.1109/ISCS61804.2024.10581226.
- [8] S. Jagan, R. Pokhariyal, K. Mahajan, C. L. N. Deepika, P. D. Sudha and A. Dutta, "Machine Learning with Deep Learning Approach for Cyber Security Threats Prevention Model," 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICES), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICES60034.2023.10465570.