INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL

Machine Learning to Improve Security Operations Centers

NORBERTO JOÃO GOMES LOPES DE SOUSA

julho de 2022

P.PORTO

MEIA
MESTRADO ENGENHARIA
INTELIGÊNCIA ARTIFICIAL

Machine Learning to Improve Security Operations Centers

Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Porto School of Engineering

2021/2022

Norberto João Gomes Lopes de Sousa

1120608

Thesis Jury

President:

Dr. Luiz Felipe Rocha de Faria

Coordinator Professor, Polytechnic of Porto - School of Engineering

Vocals:

Dr. António Alberto dos Santos Pinto

Coordinator Professor, Porto School of Management and Technology

Dr. Isabel Cecília Correia da Silva Praça Gomes Pereira

Coordinator Professor, Polytechnic of Porto - School of Engineering

isep Instituto Superior de Engenharia do Porto

Porto, June 2022

Machine Learning to Improve Security Operations Centers

Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Porto School of Engineering

2021/2022

Thesis submitted for the

Master's Degree in Artificial Intelligence Engineering

authorship of

Norberto João Gomes Lopes de Sousa

1120608

supervised by

Prof. Dr. Isabel Cecília Correia da Silva Praça Gomes Pereira

and co-supervised by

Prof. Orlando Jorge Coelho De Moura Sousa

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Porto School of Engineering (ISEP) for being my home in the past two years, working for its research department Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD) and studying for this Master's Degree in Artificial Intelligence Engineering.

To my supervisors Prof. Isabel Praça, Prof. Orlando Sousa and Eva Maia, I thank you for your help throughout this thesis and appreciate all the guidance you have given me.

Finally to my friends, family and girlfriend thank you for all the patience throughout this journey.

ABSTRACT

Since the onset of the internet, the world has embraced this new technology and used it to collectively advance Humanity. Companies have followed the trend from the physical to the digital world, taking with them all their associated value. In order to safeguard this value, security needed to evolve, with enterprises employing departments of highly trained professionals. Nevertheless, the ever increasing amount of information in need of evaluation by these professionals requires the deployment of automation techniques, aiding in data analysis and bulk task processing, to reduce detection time and as such improve mitigation. This work proposes a novel tool designed to help in attack detection and alert aggregation, by leveraging machine learning techniques. The proposed solution is described in full and showcased using real data from an example implementation.

Key-Words: Cybersecurity; Machine Learning; Security Operations Center

RESUMO

Desde o aparecimento da internet, esta nova tecnologia tem sido usada para avançar a Humanidade. O mercado seguiu as tendências, passando do mundo físico para o digital e levando consigo todo o seu valor associado. De forma a salvaguardar este valor, a segurança precisou de se adaptar, com empresas a dedicarem departamentos inteiros com esse objetivo. No entanto, a quantidade cada vez mais elevada de informação a analisar exige o desenvolvimento de técnicas automáticas de processamento de dados e execução de tarefas em massa, para diminuir o tempo de deteção de ataques permitindo uma mitigação mais ágil dos mesmos. Este trabalho propõe uma ferramenta projetada para ajudar na deteção de ataques e agregação de alertas, usando técnicas de inteligência artificial. A solução proposta é descrita na íntegra e apresentada usando dados reais aplicados a uma implementação de exemplo.

Palavras-Chave: Cybersegurança; Inteligência Artificial, Centro de Operações de Segurança

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
ICT	Information and Communications Technology
IDS	Intrusion Detection System
IoC	Indicator of Compromise
IoT	Internet of Things
KNN	K-Nearest Neighbors
MISP	Malware Information Sharing Platform
ML	Machine Learning
PoC	Proof of Concept
SIEM	Security Incident and Event Management
SOAR	Security Orchestration, Automation and Response
SOC	Security Operations Centers
SVD	Singular Value Decomposition
SVM	Support Vector Machines

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Ever since the dawn of Mankind, value has both been protected and attacked. Throughout History, this value has evolved and taken many forms, as have the mechanisms used to secure it. From the humble vigilance of a tribe's sustenance against thieves and animals, to the ingenious and constant development of vaults, warding physical breaches [1], this permanent game of cat-and-mouse has followed Humanity from the beginning. Now, with the onset of the internet not only impacting current industries but creating new ones as well, security has needed to evolve in order to protect these online value chains [2].

The internet allows cybercriminals to treat their craft as a business, always testing and improving innovative ways of circumventing cybersecurity and maximizing profits [3]. The fast pace with which things are developed in this current reality, means the typical Reactive Security is no longer enough [4]. Although this type of Security is a necessary baseline for any enterprise, involving the deployment of antivirus and firewalls, it hinges on a key component: previous attacks [5]. In this, measures are taken to be certain that known attacks are prevented or stopped when they occur. The issue arises once a new type of attack is created or a new type of vulnerability is discovered, and a successful exploit is launched before it is patched. With data breaches reaching hundreds of thousands in damages per incident, lessons quickly become a costly affair [6]. As such, security needs to be paired with more proactive methods of staying ahead of the curve, such as up-to-date security training and sensibilization for personnel as well as regular systems vulnerability tests.

Security Operations Centers (SOC) normally orchestrate these tasks, along with the gathering of information from normal business operations as well as status reports from all used devices and network systems [7]. A security analyst will then have the responsibility of reviewing all these logs and acting on suspicious behaviour. However, with thousands of logs reviewed every day, tools need to be created to aid in this work, by filtering and correlating data, and highlighting important discoveries to analysts. The use of Artificial Intelligence (AI) in problems of this type has achieved promising results [8].

Furthermore, similarly to cybercriminals' sharing of unpatched exploits [9], collaboration in cybersecurity is also an important step to levelling the playing field. Projects such as Malware Information Sharing Platform (MISP) have proven a very effective tool by cataloguing all information on known attacks and exploits. This ensures proper solutions to any security breach are developed and disseminated throughout the community, reducing the impact of even zero-day attacks [10].

1.2 Objectives

This work has the main purpose of applying automation to cybersecurity tasks in order to more efficiently deal with new threats, for that several targets have been appointed to better guide this thesis to that end. They are as follows:

- **O1:** Identify the main problems that cybersecurity currently deals with.
- **O2:** Study the state of the art of techniques and technologies utilized in this domain.
- **O3:** Proposed a solution to the identified problems.
- **O4:** Deploy an example implementation, highlighting the obtained results.

1.3 Research Questions

The following Research Questions were used to better guide the investigations performed in this thesis:

- **RQ1:** How can Automation help in day-to-day cybersecurity tasks?
- **RQ2:** What are the best techniques for each type of task in a SOC?

In **RQ1** SOCs and their main tasks are analysed, to better understand just how much reliable automation can benefit all involved. Additionally, in **RQ2**, Machine Learning is explored as a possible answer to alleviate SOC analyst's workload of monotonous and repetitive tasks.

1.4 Contributions

From this work, several contributions can be appointed:

- **C1:** A survey on the state of cybersecurity tools deployments and currently used automation techniques.
- **C2:** Guidelines for deploying a novel solution to some of the SOCs more tedious tasks.
- **C3:** An example implementation of the proposed solution with real world data.

1.5 Outline

This work is organized into multiple sections, described as follows:

- Chapter 1 introduces the basic concepts explored throughout this work as well as the research questions that will be addressed.
- Chapter 2 describes the motivations behind this work, going more in-depth on each of the related concepts and exploring the relationships between them.
- Chapter 3 contains an overview of the current state of technology in the main topics of this work.
- Chapter 4 delineates the proposed solution, explaining the different parts and how they fit together.
- Chapter 5 applies the solution to a given case, outlining and comparing the obtained results, as well as detailing the technical infrastructure.
- Chapter 6 discusses the overall findings of this work as well as highlighting possible future work.

CHAPTER 2

MOTIVATION

As cyberthreats are becoming increasingly common, it is crucial to react quickly to security breaches in order to limit an attackers impact in the network [11]. A SOC is a centralized place for safeguarding an organization's status, by establishing and maintaining Situational Awareness of its security [7]. Situational Awareness means that a SOC understands its overseen environment, by knowing the details of all events happening regarding its Information and Communications Technology (ICT) devices and understands the existing threats to them [12].

Three main sections can be considered in Situational Awareness: Environment, Mission and Threat. Environment encompasses all hardware and software utilized by the organization; the location, type and quantity as well as all connections between them, including possible vulnerabilities. Mission and Threat, on the other hand, are less introspective, standing side-by-side in paying attention to the surrounding environment instead. Mission regards the business side of the organization and how it interacts with its partners, and Threat considers possible enemies and what they stand to gain in an attack.

2.1 The Flood of Data

In order to build the Environment part of Situational Awareness, SOCs collect log data from a myriad of devices, servers, workstations, firewalls, routers, *etc.*, for every action involving each of them. When a simple email is sent, logs are generated from every single device that interacts with the email, from the workstation used to send it, to the email server and the network devices responsible for carrying it. Even moderately sized organizations generate huge amounts of data [7]. Correlating data from different sources helps build a much clearer picture, by making it possible to follow the series of events, and as such detect intrusions in a timely manner. Correlation involves comparing logs from the different sources and connecting them in a timeline.

After collection, SOC use tools to correlate data and launch alerts with the findings to be reviewed by analysts. Most correlation engines in these tools are rule based, were rules are developed by security experts and compiled in sets, in order to defeat a type of attack. Developing these rules is a time consuming endeavor requiring a high level of expertise and precision. Furthermore, as these rules reflect the current threat landscape and threats evolve overtime, new rules need to be continually developed and tested in an iterative process that leaves gaps in the security fabric of an Organization [13].

Nowadays, many of the challenges of cybersecurity can be attributed to the flood of data and repetitive tasks within the workflow of a SOC whittling down the available time of security experts. Automation appears the obvious answer to this problem and, although certain tasks do require a more hands-on approach to them, such as devising and maintaining rule-based systems, others benefit greatly from it, such as threat identification [14]. For this latter one, with its bulk data processing nature, application of Machine Learning (ML) techniques actually benefits from a high amount of data to compile [15].

2.2 Collaboration in Cybersecurity

The Threat component of Situational Awareness considers an Organization's value and significance, as that indicates what types of attacks it will be submitted to. This information can be critical when devising a cybersecurity strategy as a script-kiddie's motivation and ability is vastly different from state-sponsored cyber-terrorism, with vastly different outcomes [16, 17].

Mission, on the other hand, identifies what vulnerabilities allies introduce. The implicit trust relationship existent between business partners can be abused to attack an Organization, either from access to the environment from a trusted network or from targeted phishing from known emails. Furthermore, attacks to an Organization higher up in the supply chain can have impacts on all others downstream, *i.e.*, attacking the supplier of the intended target or targets [18, 19], or infecting production environments by targeting open source libraries [20, 21]. These types of attacks, although lengthy in their execution, can completely bypass an otherwise robust security strategy.

Linking SOCs from different Organizations, preferably within the supply chain, in a collaborative security effort has many advantages, among which strengthening against supply chain attacks. Besides setting mutual objectives in security strategies and as such, presenting a common front against cyberattacks, this collaboration further allows the sharing of expertise among analysts while dividing the load dealing with alerts. Nevertheless, access to a whole new SOC worth of assets also means a SOC worth of information to go through, exacerbating the issue of too much data.

2.3 Workflow Automation

The typical SOC workflow 2.1 can be simplified into two main parts, Threat Management and Incident Management. Threat Management involves monitoring an environment through analysis of incoming information, in order to identify possible threats in a timely manner. Incident Management, on the other hand, involves dealing with the detected threats either through mitigation or remediation.

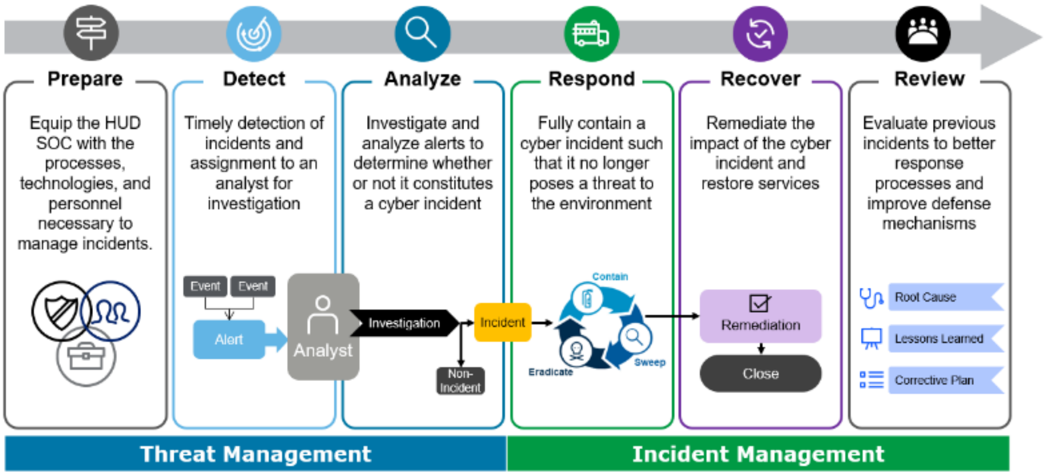

Prepare
Equip the HUD SOC with the processes, technologies, and personnel necessary to manage incidents.
Detect
Timely detection of incidents and assignment to an analyst for investigation
Analyze
Investigate and analyze alerts to determine whether or not it constitutes a cyber incident
Respond
Fully contain a cyber incident such that it no longer poses a threat to the environment
Recover
Remediate the impact of the cyber incident and restore services
Review
Evaluate previous incidents to better response processes and improve defense mechanisms
Event
Event
Alert
Analyst
Investigation
Incident
Non-Incident
Contain
Eradicate
Sweep
Remediation
Close
Root Cause
Lessons Learned
Corrective Plan
Threat Management
Incident Management

Figure 2.1: SOC workflow [22].

As information moves through the SOC workflow, its designation is upgraded according to its threat status, Figure 2.2. Individual device logs are Events, a simple occurrence originating somewhere in the environment that by themselves may or may not indicate suspicious behaviour. As previously mentioned, multiple Events are correlated together using semi-automatic methods such as rule based systems [13], emitting an Alert when a combination of Events triggers a given rule set. Alerts can be false positives, and as such inconsequential, or true positives (Incidents) meaning a cyberattack is occurring and mitigation or remediation actions need to be deployed.

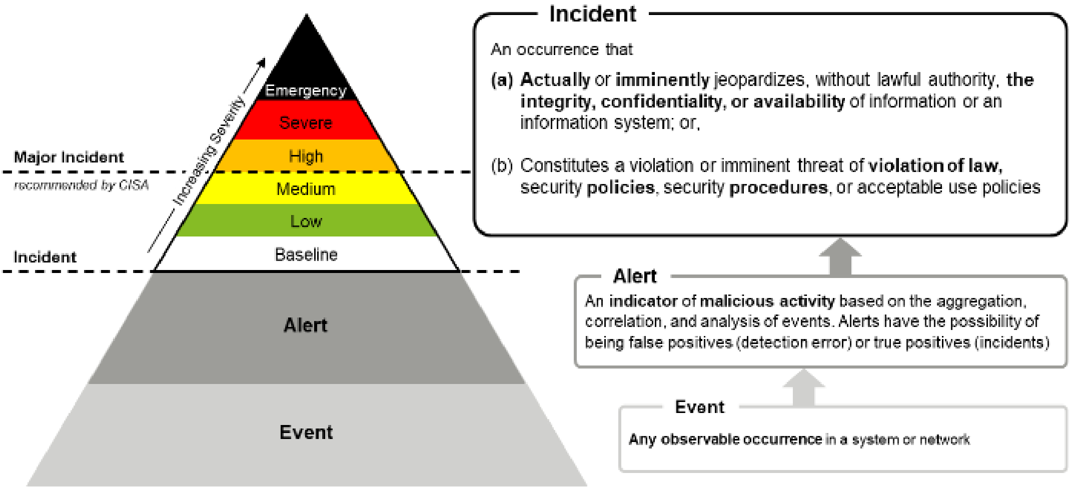

Major Incident
recommended by CISA
Incident
Increasing Severity
Emergency
Severe
High
Medium
Low
Baseline
Alert
Event
Incident
An occurrence that
(a) Actually or imminently jeopardizes, without lawful authority, the integrity, confidentiality, or availability of information or an information system; or,
(b) Constitutes a violation or imminent threat of violation of law, security policies, security procedures, or acceptable use policies
Alert
An indicator of malicious activity based on the aggregation, correlation, and analysis of events. Alerts have the possibility of being false positives (detection error) or true positives (incidents)
Event
Any observable occurrence in a system or network

Figure 2.2: Incident Types [22].

For Incident Management, its mitigation and remediation tasks can be somewhat automated through the use of playbooks [23]. Security Experts will prepare playbooks, describing step-by-step how to deal with a given type of intrusion or attack. A phishing email playbook might include steps such as: deleting emails from affected inboxes, sending out memos raising awareness about this type of attack and running scans on the machines of victims to make sure nothing was compromised. Compiling a list of playbooks for different types of attacks allows automation of much of their steps, considerably streamlining a SOCs workflow [23]. Furthermore, multiple alerts originating from the same type of attack, or even the same attack, can be aggregated in cases, where playbooks can be applied to all the alerts in a case at the same time.

The problem of too much data impacts Threat Management more heavily, with detection rules requiring continuous maintenance and alerts needing to be manually analysed in an individual manner. Detection and Analysis becomes then the most time consuming tasks, for after pinpointing the correct type of attack, chances are that a playbook with automated tasks already exists. For this reason, in the hopes of further automating cybersecurity, this detection step is continuously analysed in the literature [24, 25, 26, 27].

2.4 Summary

When devising a security strategy there are several critical steps to consider. It is not just the latest tools, or the most knowledgeable personnel that create the most robust system, but the complete strategy where all parts are collaborating. Security staff needs the correct tools to help them with the workload, as well as the right processes to streamline interventions. The tools need to be configured correctly alongside the devices being monitored. Finally, with the advances in machine learning, new automation techniques become available, allowing even greater efficiency.

CHAPTER 3

STATE OF THE ART

Although the training and experience of a SOC's staff has huge value, they need to be supported by the correct tools to properly perform their tasks. Truly efficient SOCs utilize information automation in combination with their analysts' expertise, to streamline data filtering and correlation processes.

3.1 Cybersecurity Tools

Security Incident and Event Management (SIEM) systems are powerful tools that combine log management systems with several security related features such as data analysis and correlation, as well as rule based alerting. Collected data is sent to a centralized unit and used to build a baseline of normal behaviour for an Organization [28]. Beyond regular business operations, user authentication and authorization is also monitored taking into account privileged users and sensitive data, as required by compliance reporting [29]. Continually updated threat intelligence is used to warn of any weaknesses in applications or systems in the environment, that might be exploited by targeted attacks. Finally, data analytics functionalities are capable of communicating meaningful patterns through dashboards and reports, aiding in security investigations.

These systems however, can still generate thousands of alerts that normally are monitored using mostly manual methods and processes [30].

Security Orchestration, Automation and Response (SOAR) are possible solutions to problems with manual threat analysis, delays in dealing with intrusions as well as providing continuous, centralized and up-to-date security status of an Organization. SOARs' first step is *Integration*, where all the tools in a SOC are centralized in a single platform, increasing their interoperability. Most security tools offer API-based unification support [30], but as different tools generate data in different formats, so do these formats need to be standardized. The second step is *Orchestration*, where security experts create playbooks of their security tasks, by developing code or scrips that interact with the correct tools for each case, facilitating investigations or responses to security incidents. The third and final step is *Automation*, where an Organization identifies which processes can be automated and which require review by security analysts. This step varies greatly from one Organization to the next, as there is no clear consensus on what tasks are more critical.

In a Forrester Wave report on security platforms [31], J. Blankenship *et al.* rated and described some of the market leaders in this domain, as seen in Figure 3.1. In addition, other recent vendors have entered the market with robust and interesting security tools:

- Splunk [32]: Splunk is a SIEM solution able to be hosted in the cloud and interface seamlessly with mature environments, while also being available in a Software-as-a-Service (SaaS) subscription model. It provides the normal SIEM functionalities such as data ingestion from most sources, multiple types of correlation engines and intuitive data visualization to aid in security investigations.

- Phantom [33]: Splunk also offers a SOAR solution, Phantom. Providing all the most important functionalities of SOARs, security infrastructure orchestration, playbook automation, case management capabilities and integrated threat intelligence.

- Cymerius [34]: this tool is an all-in-one SIEM and SOAR, built for Critical Systems with multiple locations. It is therefore equipped with powerful data ingestion and correlation capabilities to a unified interface, as well as allowing the orchestration of tools and humans while respecting the standards and procedures of an Organization.

- IBM Security [35]: IBM is building a comprehensive security platform in the cloud, capable of covering all security needs. With SIEM and SOAR capabilities, as well as offering supplementary functionalities available in the app format, allowing an Organization to pick and choose the most useful tools for their environment.

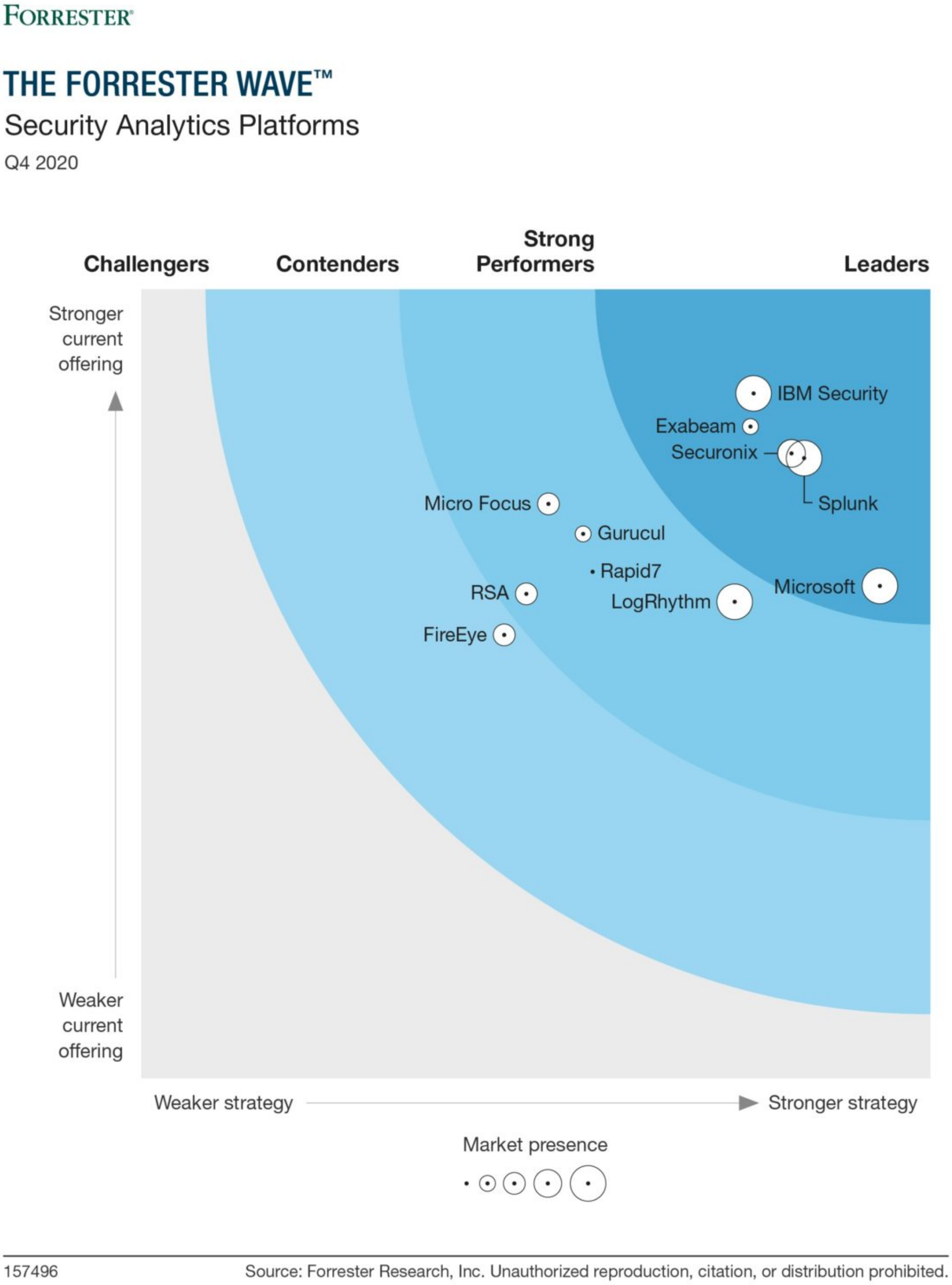


Figure 3.1: Security Platforms Market Leaders [31].

- Microsoft Sentinel [36]: Microsoft offers a cloud-based security analytics platform, a SIEM with great compatibility with the other Microsoft products such as Azure, Office365 and Windows Defender, allowing great integration and protection at multiple layers, from email to cloud and endpoints.

- The Hive [37]: is an easily deployable, open-source security orchestration platform, designed with collaboration in mind. It offers all the required functionalities of a SOAR for multiple partners, with the ability of sharing data, security playbooks and processes, as well as allowing analysts from different organizations to work collaboratively on the same cases.

Despite the existence of multiple commercial solutions available for SIEMs, SOARs and Incident Management portals the research community has not stopped developing new and innovative ways to improve these tools.

3.2 Machine Learning

Machine Learning encompasses a vast number of different techniques, based on a multitude of training strategies and computing algorithms. From supervised and unsupervised to hybrid and deep learning, new techniques are continuously being developed and tested in the cybersecurity domain [38]. Due to the nature of security tasks such as data correlation and intrusion detection, it is no surprise that ML achieves good results [39].

Supervised learning, where labeled training data is used to train the models, has seen many uses in applications such as Intrusion Detection Systems (IDS), analysing incoming data and detecting attacks. In [40], Ł. Podlodowski *et al.*, applied a XgBoost classifier to a suspicious network event dataset, demonstrating its great performance in classification problems.

Random Forest [26] and Support Vector Machines (SVM) [27, 26] are also represented with good results, although their applications are certainly pivoting to ensemble implementations. In [26], A. Yeboah-Ofori *et al.* explores several models in this domain, comparing their results for threat detection, while in [27] S. Kalyani *et al.* applies SVM directly to security auditing, discussing the possibility of application to different systems.

Unsupervised approaches such as the ones based on anomaly detection are also being investigated in the literature. By modeling the network during normal operations, deviations from that normal baseline will standout, possibly detecting zero-day attacks.

In [41], K. Ghanem *et al.*, have enriched an existing IDS system by employing a Support Vector Machines based, anomaly detection model. In a similar manner, A. Siddiqui *et al.* in

[42] developed a cyber attack detector based on anomaly detection, with explainable results and capable of improving using the feedback of security experts.

In [27], M. Evangelou *et al.* defends anomaly based systems as unbeatable secondary defense systems, exploring several models for this function and finding Quantile Regression Forests the ideal model for the used environment of enterprise networks.

Hybrid ML approaches are being used as a means of improving the results of single classifiers, in threat detection. In [43], T. Dias *et al.* presented a hybrid IDS solution that used ML to enrich expert written rules. Similarly, in [44], M. Aydın *et al.* combines missuse based and anomaly detection based IDS systems to achieve a different type of hybrid IDS, testing this solution in MIT Lincoln Laboratories' network traffic data [45].

In [46], P. Shukla *et al.* tested several ML based IDS systems applied to Internet of Things (IoT) security, comparing supervised and unsupervised based approaches to an hybrid one using both types of models, with the hybrid approach achieving better results overall with less false positives.

In [47], J. Carneiro *et al.* compared several ML models applied to network based intrusion detection, using the notable CIDDS-001 dataset [48]. Similarly, N. Oliveira *et al.*, applies several ML models to the CIDDS-001 dataset, focusing on how feature engineering can impact different model types.

In [49], Ployphan Sornsuwit *et al.* used an ensemble of ML models, in combination with adaptive boosting, to achieve state of the art results in multiple datasets of different types of attack. The novel model architecture in conjunction with correlation-based feature selection in the data preprocessing step, confirmed in the experimental results the higher efficiency in intrusion detection over other methods. In [50], J. Sakhnini *et al.* apply an ensemble of deeplearning methods to attack detection, focusing on attacks on the physical. The dataset used to evaluate the model is based on smart grids and developed by Oak Ridge National Laboratories.

To excel in this domain, and develop increasingly complex and useful models, data representing the day-to-day reality of a secure environment needs to be available. The closer to reality the data is, the more useful will be the resulting artificial intelligence. Its becoming common to see in the literature this problem being tackled in manners benefiting the entire research community. In [51], James Fraley *et al.* created a dataset of security alerts and a security analyst's decisions and actions for every one. Training a Neural Network with this dataset achieved 90% accuracy in security events and alerts classification. In [52], K. Highnam *et al.* present the BETH dataset, developed from baseline network honeypot behaviour and designed for anomaly detection model training.

Although ML shows promise in cybersecurity, it can have disadvantages by introducing new attack vectors in a system. Adversarial AI Attacks can have devastating consequences, by allowing attackers to completely bypass an Organization's security [53]. These types of attacks feed slightly altered information to a model during retraining phases, in order to manipulate it into wrongly classifying malicious behaviour as benign [54]. Steps are being taken to research this type of attack though, with G. Apruzzese *et al.* in [55] discusses how many commonly used machine learning models in cybersecurity are vulnerable to these types of attacks, suggesting possible fixes for current models and improvements to avoid these vulnerabilities going forward. In [24], the same authors apply adversarial attack training to random forest model implementations to great success, not only protecting the models from these types of attacks but also improving performance.

3.3 Summary

With security tools being continually developed, their focus is on how to be more efficient with analysts time allowing them to be more productive. Automation seems the clear solution, with machine learning the promising new technology.

Despite the extensive exploration of ML applied to cybersecurity, the focus is undoubtedly on attack detection, leaving a clear hole in other promising automatable steps, namely alert group aggregation. Going forward, a robust security strategy will be a balancing act of human experience and expertise, allied with ML-powered automation.

CHAPTER 4

CONCEPT

Automation and the application of machine learning are a promising solution to mitigating the flood of data in cybersecurity. Although the use of AI is not a new endeavour in this domain, the literature currently mostly focuses on the analysis of incoming data in order to identify a security breach. Albeit a commendable use, other steps in a SOC workflow could greatly benefit from the employment of this technology.

Current SOC implementations are typically setup so that alerts from one or multiple sources are delivered straight to the used incident management solution 4.1, where analysts have access to alerts in a queue similar to a ticketing system [56]. Analysts then manually access alerts individually for signs of security breaches.

In order to further automate cybersecurity, and focusing on the identification step of a SOC workflow, this work proposes a solution capable of slotting into current SOC tools, as a way of enriching incoming alerts. This solution is leveraged as a decision support system, employing multiple models to perform identification and classification of alerts, adding their results as another point of consideration for security expert analysis. The additional information helps analysts not only decide if a given alert is in fact an attack, but also by identifying which case contains playbooks to treat similar alerts.

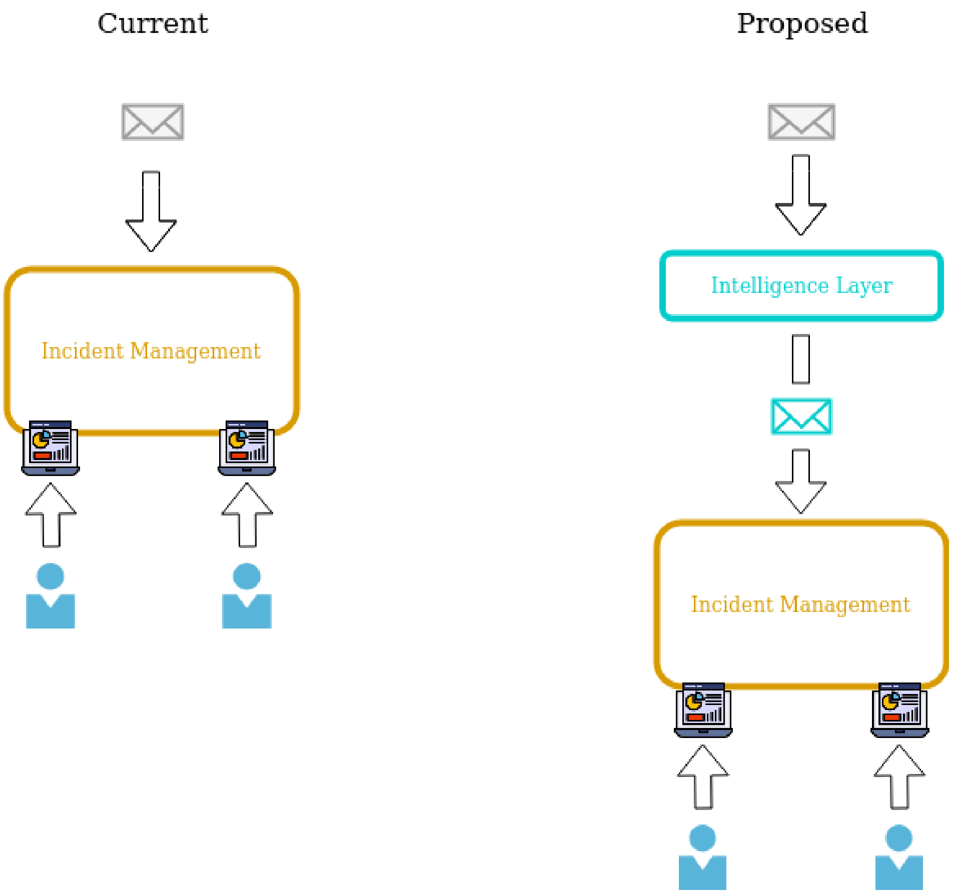


Figure 4.1: Simplified Comparison of Current vs Proposed Solutions.

4.1 Intelligence Layer Architecture

The Intelligence Layer was designed using a microservices architecture design pattern, separating responsibilities into standalone components. These components can be seen in Figure 4.2 and are described as follows:

- **Preprocessing** handles incoming alerts, applying data transformation logic in order to prepare them for model ingestion. Additionally, new features are computed using historical information from previous recent alerts.
- **Database** stores all the alerts that pass through the system. This data can be revisited and used to enrich alerts in preparation for ML analysis.
- **Analyser** contains the logic behind this system, receiving alerts after their preprocessing and submitting them to the Machine Learning Engine for analysis. After obtaining the results, this component adds the new information to the alert in a readable manner before sending it to the SOAR of choice.

- **Machine Learning Engine**, possibly the most important component, contains the models trained using relevant security data and capable of not only detecting an alert as a security breach but also aggregating it to other similar alerts in the system, for bulk processing by security analysts.

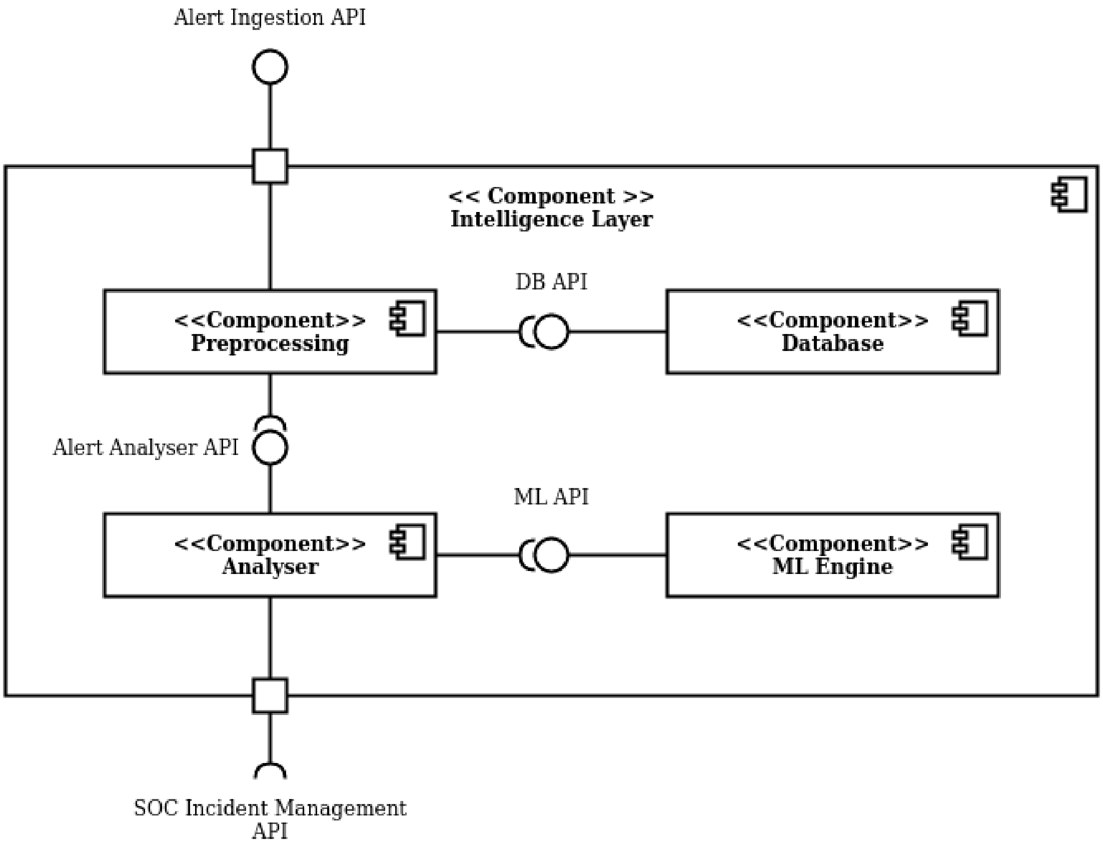


Figure 4.2: Decision Support System: Components View.

4.2 Machine Learning Engine

The ML Engine aims to tackle two problems of the SOC pipeline, assessment of incoming alerts for security threats, or identification, and grouping of similar alerts in cases for bulk processing, or aggregation. Although different in their nature, both of these are classification problems where a set of data points are categorized into classes. In this context the data points will be alerts and the classes their possible label.

In the identification problem, only two possible classes exist for an alert is either *attack* or *normal*, whereas in the aggregation problem the possible classes are the existent cases in the system. Furthermore, the nature of the data for our binary classification problem, identification, guarantees that all the future incoming entries will only ever be of two possible types. On the other hand, classifying each alert into groups will fail when a never before seen alert, *i.e.* from a new type of attack, arrives in the queue. The multiclass classification model, trained with known classes will incorrectly identify the new alert as one of the existing classes.

For this reason, a middle step needs to exist between both classification problems. After being identified as *attack* by the first model, the system needs to decide if this alert is similar to other alerts already in the database or if it is a new one. As such, an anomaly detection model will be trained with alerts already in the system to create a baseline of known alerts, filtering any outliers and skipping the final step. The third model is trained on groups of alerts that compose a case, selecting the relevant case for every incoming entry. The sequence of these three steps can be seen in Figure 4.3

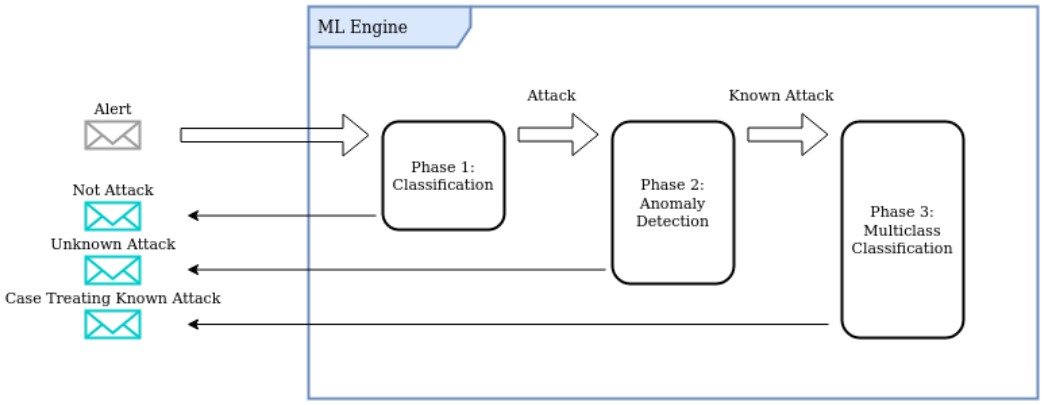

ML Engine
Alert
Not Attack
Unknown Attack
Case Treating Known Attack
Attack
Known Attack
Phase 1: Classification
Phase 2: Anomaly Detection
Phase 3: Multiclass Classification

Figure 4.3: Machine Learning Steps.

Each of the three different identification phases of the Intelligence Layer, require ML models attuned to the unique specifications of their given problem. These models will undergo a selection stage were data originating from the final system is used to train and compare the results among them.

4.2.1 Phase 1: Classification

For this problem three main models were considered, given their results with similar problems in this or other domains.

- **Random Forest** [57] is a tree based model, employing a set of decision trees and taking in account the output of each one. A decision tree aggregates datapoints by iteratively splitting the features of a given dataset into consecutive binary nodes, ending each branch on its outcome, or label. Although very good with low complexity data, higher sized trees can lead to overfitting. Random Forest models mitigate this issue by using an ensemble of unrelated decision trees and consolidating their results, achieving significant results in the literature for both classification and regression problems.

- **Support-Vector Machines (SVM)** [58] is a probabilistic model that maps training data to points in space, and finds the hyperplane with the maximum margin that separates the two classes. Newer data points are mapped in space in the same way and classified according to which side of the hyperplane they have landed. This model is a very robust classifier with the caveat that it is limited to binary-class classification.

- Similarly to Random Forest, **XgBoost** [59] is an ensemble of decision trees, but using a gradient boosting algorithm. Instead of concurrently training a group of decision tree models and averaging their output, models are trained consecutively using the residuals from each iteration to train the next one.

4.2.2 Phase 2: Anomaly Detection

In order to classify an incoming alert as ”unknown”, an anomaly detection based approach was selected. Although this approach is not uncommon for the cybersecurity domain, it is normally applied to the detection of attacks. Here, its use is to identify alerts different from everything in the system so that new cases can be aggregated and playbooks developed. For this novel use case, two models were selected.

- **Isolation Forest** [60] is a tree based model that uses distance between data points to detect outliers, hinging on the principle that outliers are distinct from normal data. During the construction of the binary tree, data is grouped into branches according to their similarity, with more similar entries needing longer branches to differentiate them. As such, data closer to the root of the tree can be considered an anomaly since it was easily distinguishable from the rest.

- **One-Class Support-Vector Machines** [61] is a similar implementation to SVM but instead of using an hyperplane to separate two classes, it uses an hypersphere around normal data and classifies new data based on its distance to the sphere.

4.2.3 Phase 3: Multiclass Classification

Since multiclass classification is a subset of normal classification, they have in common many models that achieve similar results. For this reason some of the models from the first phase were selected:

- **Random Forest** due to its robust results and straightforward implementation, behaving no differently in binary and multiclass classification problems.
- Although models such as **Support-Vector Machines** in its most simple type only supports binary classification, implementations exist where the problem is compartmentalized into multiple binary classification problems followed by the same principle: discovering the hyperplane that linearly separates classes [62, 63].
- **K-Nearest Neighbors (KNN)** [64] uses distance between datapoints to identify clusters of similar data. Despite its good results it is not very scalable due to being computationally demanding.

4.3 Summary

The proposed solution allows slotting into current SOC workflow implementations, enriching alerts with valuable information. Several ML models are presented for three different phases, responsible for identifying attacks and aggregating them into the correct cases.

CHAPTER 5

EXPERIMENTAL FINDINGS

For the proposed solution to work, the three used ML models need to be trained, tested and deployed in the system. As a case study for this, a dataset was acquired and all the relevant steps of constructing usable models are described and analysed, and the system deployment into a SOC workflow is documented. The used SOC is based on the Security of Air Transport Infrastructure of Europe (SATIE) Project [65], where new SOC philosophies were being investigated, with new intelligent tools and innovative workflows being explored and developed.

5.1 Datasets

The data used is sourced from SATIE's cyber-physical environment and composed of two parts. The first one is of alerts, correlated from logs from a multitude of devices and device types, from network devices such as routers and firewalls, to workstations and even physical sensors. The second is a list of incidents, meaning a security breach confirmed by a specialist, where each incident contains the list of alerts related to it. Using this information, a dataset can be built from the alerts using the list of incidents to label each one. All alerts related to an incident will receive that incident's ID number as label, with the rest being marked as ”normal”.

Additionally, a new set of features were engineered, aiming to give models historical context for each entry. These features were extracted using a rolling time window and are based on the recent frequency of incoming alerts. As such, for each entry and given a window of 30 minutes, the number of alerts received, the number of distinct sources and the most common source are computed and added to the dataset.

Due to the difference in log sources, each alert contains differing information. This results in a dataset resembling a sparse matrix, with many entries having absent values in unrelated features. In order to reduce the impact this could have in the final results, three different datasets were constructed and tested. The first one, henceforth referred to as Full Dataset, contains all the data available, serving as the control sample and its features can be seen in Table 5.1. The second dataset went through a curating process where columns were disregarded if containing a threshold of missing values higher than 60%, from now on referenced as Curated Dataset. Finally, for the third dataset, a Random Forest was trained for a classification problem and the most important features derived from that training were selected to build the Feature Importance Dataset. These last two datasets' features can be seen in Table 5.2 and 5.3 respectively.

In Phase 1, for the classification problem, 11764 total entries are used, with 11201 labelled as ”normal” and 563 labelled as ”attack”. For the following phases, ”normal” labelled entries are no longer relevant and as such are disregarded. In Phase 2, for the anomaly detection problem, 166 outliers were randomly selected, leaving 397 as inliers. In Phase 3, for the multiclass classification problem, of the 563 ”attack” alerts, each was aggregated into classes according to its incident, ranging from 3 to 100 members each.

5.2 Data Preprocessing

Before starting the experiments some processing of the data needs to occur, to prepare it to be ingested by the ML models. Categorical data needs to be encoded, since these models do not compute data that is not numerical, but even previously numerical data needs adjustments depending on which type of models will be used later on.

When encoding data, several approaches are discussed in the literature, each with their own pros and cons, but arguably the most used is OneHot Encoding [66]. In this type of encoding, a categorical feature of *n* unique values is unfolded into *n* binary features, each entry will be marked with ”1” for the feature pertaining to its previous one, and with ”0” for all the others. This has the caveat of considerably increasing the number of features in a dataset, but without creating undesirable numerical relationships between categorical vales. Furthermore, the prolific use of distance based models in this work, required the scaling of numerical data, as distance

Table 5.1: Full Dataset Features.

Features	Description
alert_title	Title
location	Physical location of the source
most_common_sensor_in_window	Most common source in the last 30 min
n_alerts_in_window	Number of alerts in the last 30 min
n_distinct_sensors_in_window	Number of sources of alerts in the last 30 min
sensor	Source of the alert
severity	Level of priority for this alert
type	Origin type of the alert, either cyber or physical
target_ip_address	Target IP address
source_ip_address	Source IP address
source_host_name	Host name of source machine
source_port	Source port
source_user_name	Source Username
target_host_name	Host name of target machine
target_port	Target port
source_tool	Source Tool
source_url	Targeted URL
target_file_name	Target file name
target_tool	Target tool
target_user_name	User name of target
command	Command that triggered the alert
file_name	File contained in the alert
file_hash	Hash of file contained in the alert

Table 5.2: Curated Dataset Features.

Features	Description
alert_title	Title
location	Physical location of the source
most_common_sensor_in_window	Most common source in the last 30 min
n_alerts_in_window	Number of alerts in the last 30 min
n_distinct_sensors_in_window	Number of sources of alerts in the last 30 min
sensor	source
severity	Level of priority for this alert
Type	Origin type of the alert, either cyber or physical
target_ip_address	Target IP address
source_ip_address	Source IP address

Table 5.3: Feature Importance Dataset Features.

Features	Description
alert_title	Title
location	Physical location of the source
most_common_sensor_in_window	Most common source in the last 30 min
n_alerts_in_window	Number of alerts in the last 30 min
n_distinct_sensors_in_window	Number of sources of alerts in the last 30 min
sensor	source
severity	Level of priority for this alert
Type	Origin type of the alert, either cyber or physical
target_ip_address	Target IP address
source_ip_address	Source IP address
source_host_name	Host name of source machine
source_port	Source port
source_user_name	Source Username
target_host_name	Host name of target machine
target_port	Target port

between datapoints will impact how each one is classified. This is so a change of a certain value to any feature has the same importance across the whole dataset. The chosen method for this was MinMax scaler, for keeping the shape of the overall data intact.

The final considered prepocessing step included in the experiments was a dimensionality reduction technique, using Singular Value Decomposition (SVD) [67]. Dimensionality reduction aims to reduce the number of features, computing out unneeded information while retaining the most important parts. In this context, SVD was chosen due to its compatibility with sparse feature spaces.

5.3 Experiments and Results

The selected models were trained and tested with the three compiled datasets, with experiments following a standardized script. For each dataset, every model was subjected to several experiments, differing in the used preprocessing of the data. The considered default preprocessing involves scaling and encoding of numerical and categorical features, respectively, regarding models were it is applicable to do so. Although best practices dictate, when using distance based models, that numerical data should be scaled and categorical data encoded, tree based models do not share this need. Additional considered preprocessing steps are dimensionality reduction via SVD, comparing the full range of data to smaller and richer subsets of it, and feature creation using kmeans to cluster similar data into groups.

5.3.1 Metrics

Respective to results comparison, Accuracy [68] is the base metric used for comparing the performance between models. It is the most straight forward, calculated by comparing the number of correct predictions against the total of entries predicted. Although a good baseline, it does not tell the whole story, especially in class imbalanced datasets such as the ones used in this work. Precision and Recall despite giving results that explore the data more in depth, are two sides of the same coin [69], with Precision focusing on the amount of correct entries retrieved by the model while Recall focuses on how many correct entries were identified. Consequently, F1-score [70] was selected as the metric of choice given its good balance between Precision and Recall while paying attention to class imbalance existent in the data. This imbalance can also be observed in the difference between macro and weighted metrics, since macro metrics take into account the number of each class' members during result calculations. As such, for the first and third phases, the macro F1-score was used for evaluating the impact differently sized classes have in the final results. For the second phase F1-score was also used, only this time focusing on the score for the outlier class, *i.e.* the alerts considered unknown to the system.

5.3.2 Phase 1: Classification

Experiments were conducted with the selected models for all three compiled datasets: Full, Curated and Feature Importance Datasets. In all of them, the preprocessing applied to the data consisted of scaling of numerical data using MinMax Scaler and encoding of the categorical data using OneHot Encoding. Due to the sparse nature of the datasets, an additional trial was included utilizing Dimensionality Reduction techniques, namely Singular Value Decomposition. The mapping of the experiments for each dataset is in Table 5.4 with the results in Tables 5.5, 5.6 and 5.7:

Table 5.4: Phase 1 Experiments.

	SVM	Random Forest	XgBoost
Default:	**default_svm**	**default_rf**	**default_xgb**
Default with SVD:	**svd_svm**	**svd_rf**	**svd_xgb**

Table 5.5: Phase 1 Results: Full Dataset.

Full Data	default_svm	svd_svm	default_rf	svd_rf	**default_xgb**	svd_xgb
Accuracy	0.973	0.970	0.971	0.968	0.973	0.968
Macro Precision	0.840	0.485	0.764	0.712	0.784	0.713
Weighted Precision	0.966	0.940	0.966	0.961	0.967	0.961
Macro Recall	0.584	0.500	0.653	0.617	0.661	0.624
Weighted Recall	0.973	0.970	0.971	0.968	0.973	0.968
Macro F1-Score	0.631	0.492	0.692	0.649	**0.704**	0.655
Weighted F1-Score	0.965	0.955	0.968	0.963	0.969	0.964

Table 5.6: Phase 1 Results: Curated Dataset.

Curated Data	default_svm	svd_svm	default_rf	svd_rf	**default_xgb**	svd_xgb
Accuracy	0.972	0.970	0.972	0.970	0.973	0.971
Macro Precision	0.792	0.485	0.770	0.746	0.788	0.767
Weighted Precision	0.964	0.940	0.967	0.963	0.967	0.965
Macro Recall	0.598	0.500	0.660	0.625	0.654	0.639
Weighted Recall	0.972	0.970	0.972	0.970	0.972	0.971
Macro F1-Score	0.643	0.492	0.699	0.664	**0.698**	0.681
Weighted F1-Score	0.964	0.955	0.968	0.965	0.968	0.966

Table 5.7: Phase 1 Results: Feature Importance Dataset.

Feature Importance	default_svm	svd_svm	**default_rf**	svd_rf	default_xgb	svd_xgb
Accuracy	0.984	0.984	0.984	0.983	0.983	0.983
Macro Precision	0.492	0.492	0.766	0.680	0.743	0.742
Weighted Precision	0.968	0.968	0.979	0.975	0.977	0.977
Macro Recall	0.500	0.500	0.579	0.539	0.553	0.539
Weighted Recall	0.984	0.984	0.984	0.983	0.984	0.984
Macro F1-Score	0.496	0.496	**0.621**	0.562	0.585	0.566
Weighted F1-Score	0.976	0.976	0.980	0.977	0.978	0.978

In this phase XgBoost shines, consistently outperforming the other two models. The only test where this is not true is when using the dataset constructed using feature importance. The explanation for this is quite simple, the most important features used to construct this dataset were extracted by training a random forest and sorting the features with the most impact for it. As such that dataset is tuned for use with random forest, achieving worst overall results with different models.

Although the use of automatic dimensionality reduction in the form of SVD produced worst results in most cases, the manual dimensionality reduction that happens from the Full to the Curated Dataset results in minor differences, indicating that superfluous features do exist in the full data and SVD is just not able to extract them. With more careful feature tuning the presented results can further be improved.

5.3.3 Phase 2: Anomaly Detection

In the second phase two experiments were run for each model and for each dataset, trying the default preprocessing and comparing its results to the ones with a dimensionality reduction step. Since the Feature Importance Dataset was compiled using a classification model to select the most important features, those might not reflect the best ones for anomaly detection. For this reason, this dataset was disregarded and the experiments used only the Full and Curated Datasets. Additionally, anomaly detection involves a sensibility threshold in order to classify data as inlier or outlier. As this threshold can have a great impact in the model's final results, an optimization step was added to all the experiments. This optimization used GridSearch to systematically try all the given combinations of hyperparameters, including anomaly threshold. Finally, as in Phase 1, the preprocessing consisted of MinMax Scaling and OneHot Encoding with SVD being used for dimensionality reduction. The mapping of the experiments for each dataset is in Table 5.4 with the results in Tables 5.5, 5.6 and 5.7:

Table 5.8: Phase 2 Experiments.

	SVM	Isolation Forest
Default:	**default_svm**	**default_if**
Default with optimization:	**svm_opt**	**if_opt**
Default with SVD:	**svd_svm**	**svd_if**
Default with SVD and optimization:	**svd_svm_opt**	**svd_if_opt**

Table 5.9: Phase 2 Results: Full Dataset.

Full Data	default_svm	svm_opt	svd_svm	svd_svm_opt	default_if	if_opt	svd_if	**svd_if_opt**
F1-Score	0.746	0.823	0.346	0.737	0.106	0.818	0.821	**0.828**
Macro F1-Score	0.662	0.800	0.409	0.656	0.393	0.801	0.809	0.808
Accuracy	0.683	0.803	0.415	0.676	0.528	0.803	0.809	0.809

Table 5.10: Phase 2 Results: Curated Dataset.

Curated Data	default_svm	svm_opt	svd_svm	svd_svm_opt	default_if	if_opt	svd_if	**svd_if_opt**
F1-Score	0.732	0.732	0.633	0.807	0.225	0.815	0.747	**0.820**
Macro F1-Score	0.623	0.623	0.586	0.778	0.460	0.793	0.731	0.801
Accuracy	0.655	0.655	0.591	0.782	0.563	0.795	0.732	0.803

The need for optimization inherent to anomaly detection due to error thresholds, also reveals the boost to model performance that can be extracted. Optimized models beat their default version across the board, with Isolation Forest coming ahead in both datasets. Furthermore, while SVD failed to impact the results in any significant manner, in the previous classification problem, here it managed to slightly increase the results every time.

5.3.4 Phase 3: Multiclass Classification

Once again experiments were conducted with the selected models for all three compiled datasets, following closely phase 1's preprocessing due to the similar nature of the challenge: scaling of numerical data using MinMax and encoding using OneHot. SVD was again utilized for dimensionality reduction. The mapping of the experiments for each dataset is in Table 5.11 with the results in Tables 5.12, 5.13 and 5.14:

Table 5.11: Phase 3 Experiments.

	SVM	Random Forest	KNN
Default:	**default_svm**	**default_rf**	**default_knn**
Default with SVD:	**svd_svm**	**svd_rf**	**svd_knn**

Table 5.12: Phase 3 Results: Full Dataset.

Full Data	default_svm	**svd_svm**	default_rf	svd_rf	default_knn	svd_knn
Accuracy	0.802	0.802	0.802	0.775	0.883	0.892
Macro Precision	0.558	0.578	0.582	0.507	0.515	0.505
Weighted Precision	0.792	0.787	0.785	0.778	0.832	0.853
Macro Recall	0.640	0.640	0.640	0.540	0.629	0.631
Weighted Recall	0.802	0.800	0.801	0.774	0.882	0.892
Macro F1-Score	0.581	**0.593**	0.585	0.497	0.549	0.542
Weighted F1-Score	0.784	0.783	0.778	0.762	0.854	0.868

Table 5.13: Phase 3 Results: Curated Dataset.

Curated Data	default_svm	svd_svm	default_rf	svd_rf	default_knn	**svd_knn**
Accuracy	0.811	0.810	0.792	0.784	0.901	0.909
Macro Precision	0.636	0.636	0.555	0.531	0.652	0.644
Weighted Precision	0.813	0.813	0.792	0.785	0.879	0.885
Macro Recall	0.673	0.673	0.606	0.573	0.695	0.697
Weighted Recall	0.811	0.811	0.793	0.784	0.901	0.909
Macro F1-Score	0.623	0.623	0.555	0.526	0.644	**0.649**
Weighted F1-Score	0.795	0.795	0.777	0.769	0.883	0.893

Table 5.14: Phase 3 Results: Feature Importance Dataset.

Feature Importance	default_svm	**svd_svm**	default_rf	svd_rf	default_knn	svd_knn
Accuracy	0.811	0.810	0.801	0.801	0.874	0.883
Macro Precision	0.603	0.622	0.575	0.589	0.500	0.496
Weighted Precision	0.804	0.799	0.797	0.794	0.835	0.850
Macro Recall	0.6733	0.673	0.640	0.640	0.595	0.597
Weighted Recall	0.812	0.812	0.801	0.801	0.873	0.882
Macro F1-Score	0.612	**0.624**	0.583	0.588	0.517	0.515
Weighted F1-Score	0.792	0.792	0.785	0.783	0.849	0.859

Due to both the imbalance in class members as well as the low number of overall data entries, the results for this last set of experiments are the worst so far. Furthermore, although SVM with SVD achieved the best results for two of the three datasets, KNN also with SVD achieved the best score overall in the third dataset, Curated Dataset. The slight dispersion in the results indicates that for this small sample of data no one model is best, and more data points need to be obtained before any reliable results can be produced.

5.4 Deployment

The final step before deployment is to choose which models to use for each phase. To ease the preprocessing stage of alerts before analysis, all the models chosen are from the same dataset. This way, alerts need only to receive transformation into one format to be able to be computed by all the models. Full Dataset had the highest results in three out of two phases and will then be used as the formatting formula for the incoming alerts. With this, the proposed Intelligence Layer system is ready to be deployed into a SOC workflow and start to enrich all the new alerts.

The SOC implementation chosen for testing this tool is using TheHive as an incident management and orchestration tool, with alerts originating from a Splunk IDS deployment. The correlation rules used in Splunk monitor logs from the overseen environment and launch alerts if any event triggers a rule.

TheHive is chosen as the basis for this SOC due to its open-source implementation, and collaboration focused functionalities. TheHive is designed to support multi-enterprise SOCs in a collaborative incident management and orchestration environment. This allows security analysts and experts to share information between partners and work on cases collaboratively. Furthermore, TheHive contains connections to security threat databases, namely MISP, receiving up-to-date intelligence on any new security threats.

TheHive utlizes its own concept of observables [71], stateful properties of an alert that are likely to indicate an intrusion, allowing investigations to be run on individual or groups of observables to verify their compromise level. The source IP, file hash or sender email domain are all possible observables contained in an alert, and are all information that may indicate an attack.

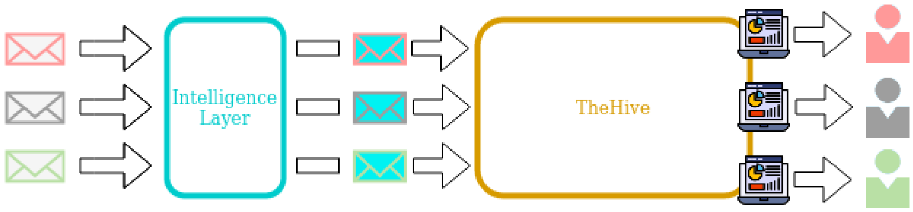
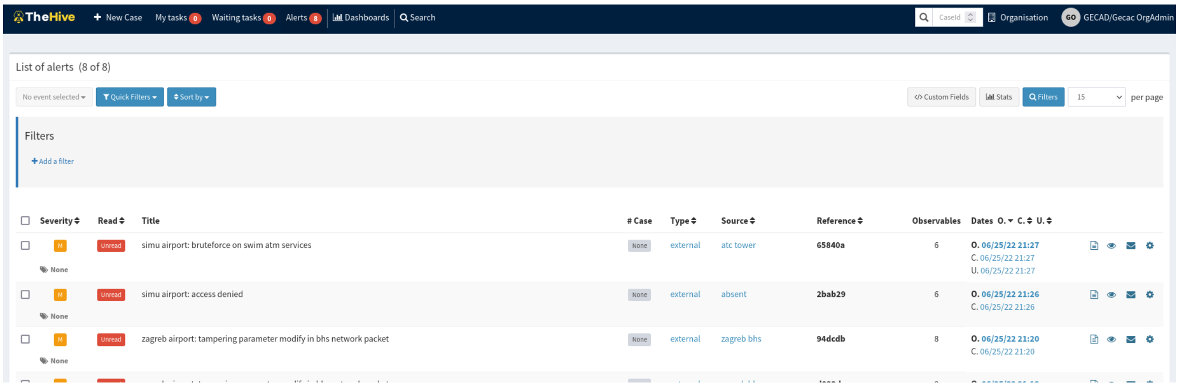


Figure 5.1: Intelligence Layer example implementation.

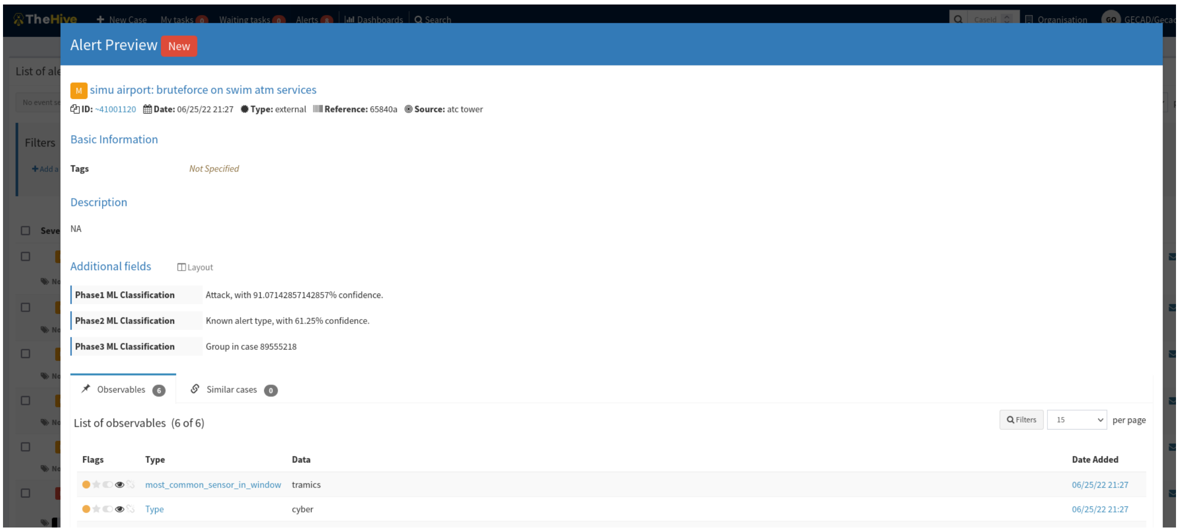
The setup differs from the one shown in Figure 4.1 due to TheHives' collaborative nature, allowing multiple enterprises to work together on the same SOC deployment, sharing information and processes when needed. As such, this implementation more closely resembles Figure 5.1, with the novel Intelligence Layer capturing incoming alerts from multiple sources and, after ML analysis, augmenting their information with intelligent classification. The improved alerts are then submitted to TheHive's new alert queue, Figure 5.2, waiting for manual verification. When security analysts log in to TheHive to perform this verification, they can use the ML analysis contained in each alert to help decide on how to proceed with each one, as seen in Figure 5.3.



List of alerts (8 of 8)

☐	Severity	Read	Title	# Case	Type	Source	Reference	Observables	Dates O. C. U.
☐	M None	Unread	simu airport: bruteforce on swim atm services	None	external	atc tower	65840a	6	O. 06/25/22 21:27 C. 06/25/22 21:27 U. 06/25/22 21:27
☐	M None	Unread	simu airport: access denied	None	external	absent	2bab29	6	O. 06/25/22 21:26 C. 06/25/22 21:26
☐	M None	Unread	zagreb airport: tampering parameter modify in bhs network packet	None	external	zagreb bhs	94dcdb	8	O. 06/25/22 21:20 C. 06/25/22 21:20

Figure 5.2: TheHive Alert Queue.



M simu airport: bruteforce on swim atm services

ID: ~41001120 Date: 06/25/22 21:27 Type: external Reference: 65840a Source: atc tower

Basic Information

Tags *Not Specified*

Description

NA

Additional fields

Phase1 ML Classification	Attack, with 91.07142857142857% confidence.
Phase2 ML Classification	Known alert type, with 61.25% confidence.
Phase3 ML Classification	Group in case 89555218

List of observables (6 of 6)

Flags	Type	Data	Date Added
●	most_common_sensor_in_window	tramics	06/25/22 21:27
●	Type	cyber	06/25/22 21:27

Figure 5.3: TheHive Alert Example.

5.5 Summary

To demonstrate the implementation of the proposed solution, an established SOC was used, with datasets compiled from the available data and used to train the ML models. Several experiments were performed to decide on the best model for each phase, with forest based models achieving the best results overall for this use case.

CHAPTER 6

CONCLUSION

This chapter discusses the overall findings of this work and highlights possible future improvements to the proposed solution.

6.1 Results Summary

Cybersecurity continually explores Machine Learning as a tool to help solve the information overload issue plaguing security centers. The novel system presented in this work aims to solve this same problem, focusing on a less explored step of a SOC workflow, aggregation of incoming alerts into cases.

The proposed system tackles this and the previous ”attack identification” step, in a robust and flexible solution, capable of integrating any current SOC tool implementation, in order to enrich incoming alerts with ML analysis of their threat risk. This additional information helps security analysts assess each alert, spending less time on each one and increasing the efficiency of the overall SOC.

As a use case of this system, a Cyberphysical dataset was acquired and used in a series of experiments, comparing possible models for the different types of ML problems existent in the system. The superiority of tree based models was obvious in the ensuing comparison, with XgBoost and Isolation Forest achieving high results in the classification and anomaly detection problems, respectively, whereas in the multiclass classification problem, despite similar results overall, KNN did edge ahead.

6.2 Objectives Overview

The objectives proposed in Chapter 1.2 were all met in the course of this work. O1 was answered in Chapter 2.1, where the main problem in cybersecurity, namely the flood of data, is described. O2 was tackled in Chapter 3, where the state of the art of cybersecurity tools and techniques are described, along the current state of Machine Learning usage in cybersecurity. O3 and O4 are addressed in Chapters 4 and 5 respectively, the former introduces design guidelines to construct a solution capable of answering the presented problem, and the latter successfully deploys a Proof of Concept (PoC) implementation on a real world example of a SOC.

6.3 Research Questions Overview

The following Research Questions were presented in Chapter 1.3 to help guide this work, with possible answers found throughout this text:

- **RQ1**: How can Automation help in day-to-day cybersecurity tasks?
  - Cybersecurity workflows in typical security centers are composed of many tedious, data heavy task benefiting from automation both in detection and analysis of attacks as well as remediation and mitigation tasks.
- **RQ2**: What are the best techniques for each type of task in a SOC?
  - While remediation and mitigation tasks benefit from straightforward automation, in the sense that similar alerts that undergo the same procedures can be grouped and treated together, threat detection and analysis requires a more nuanced approached to keep up with ever evolving attacks. For this case, ML solutions are ideal due to their capacity to evolve alongside emerging threats, learning from previous exploits and even detecting zero-day attacks.

6.4 Future Work

As future work, in regards to the use case, more data would be useful in order to improve the results across the board. Some of the limitations encountered were due to limited and imbalanced classes. On the system's side, the greatest improvement it could receive would be automatic retraining of the models, using labeled data from the SOC. Utilizing the security analysts' final decision on processed alerts to periodically improve model performance. This improvement consequently also solves the main issue with the current dataset, since with time more interesting data can be aggregated and used to improve the existing solution.

REFERENCES

[1] R. D. McCrie, "A history of security," *The handbook of security*, pp. 21–44, 2006.

[2] D. C. Wilson, "Cybersecurity Origins," in *Cybersecurity*. The MIT Press, 09 2021. [Online]. Available: https://doi.org/10.7551/mitpress/11656.003.0003

[3] A. P. Veiga, "Applications of artificial intelligence to network security," *ArXiv*, vol. abs/1803.09992, 2018.

[4] M. Huth and F. Nielson, "Static analysis for proactive security," in *Computing and Software Science*. Springer, 2019, pp. 374–392.

[5] A. Sari, M. Karay *et al.*, "Reactive data security approach and review of data security techniques in wireless networks," *International Journal of Communications, Network and System Sciences*, vol. 8, no. 13, p. 567, 2015.

[6] S. Romanosky, "Examining the costs and causes of cyber incidents," *Journal of Cybersecurity*, vol. 2, no. 2, pp. 121–135, 2016.

[7] C. Zimmerman, "Cybersecurity operations center," *The MITRE Corporation*, 2014.

[8] N. Wirkuttis and H. Klein, "Artificial intelligence in cybersecurity," *Cyber, Intelligence, and Security*, vol. 1, no. 1, pp. 103–119, 2017.

[9] V. Benjamin, B. Zhang, J. F. Nunamaker Jr, and H. Chen, "Examining hacker participation length in cybercriminal internet-relay-chat communities," *Journal of Management Information Systems*, vol. 33, no. 2, pp. 482–510, 2016.

[10] P. Rajivan and N. Cooke, "Impact of team collaboration on cybersecurity situational awareness," in *Theory and Models for Cyber Situation Awareness*. Springer, 2017, pp. 203–226.

[11] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE network*, vol. 8, no. 3, pp. 26–41, 1994.

[12] A. Evesti, T. Kanstrén, and T. Frantti, "Cybersecurity situational awareness taxonomy," in *2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, 2017, pp. 1–8.

[13] M. Cinque, D. Cotroneo, and A. Pecchia, "Challenges and directions in security information and event management (siem)," in *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2018, pp. 95–99.

[14] E. Cole, "Soc automation - deliverance or disaster," *SANS Spotlight*, 2017.

[15] E. Khramtsova, C. Hammerschmidt, S. Lagraa, and R. State, "Federated learning for cyber security: Soc collaboration for malicious url detection," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, 2020, pp. 1316–1321.

[16] R. Gandhi, A. Sharma, W. Mahoney, W. Sousan, Q. Zhu, and P. Laplante, "Dimensions of cyber-attacks: Cultural, social, economic, and political," *IEEE Technology and Society Magazine*, vol. 30, no. 1, pp. 28–38, 2011.

[17] S. Collins and S. McCombie, "Stuxnet: the emergence of a new cyber weapon and its implications," *Journal of Policing, Intelligence and Counter Terrorism*, vol. 7, no. 1, pp. 80–91, 2012.

[18] H. Boyes, "Cybersecurity and cyber-resilient supply chains," *Technology Innovation Management Review*, vol. 5, no. 4, p. 28, 2015.

[19] A. Yeboah-Ofori and S. Islam, "Cyber security threat modeling for supply chain organizational environments," *Future Internet*, vol. 11, no. 3, 2019. [Online]. Available: https://www.mdpi.com/1999-5903/11/3/63

[20] H. Zhu, "Postmortem for malicious packages published on july 12th, 2018," 2018, [Accessed June 28, 2022]. [Online]. Available: https://eslint.org/blog/2018/07/postmortem-for-malicious-package-publishes/

[21] J. Koljonen, "[cve-2019-15224] version 1.6.13 published with malicious backdoor." 2019, [Accessed June 28, 2022]. [Online]. Available: https://github.com/rest-client/rest-client/issues/713

[22] U. D. of Housing and U. Development, "Cybersecurity incident response plan," 2020, [Accessed June 28, 2022]. [Online]. Available: https://www.hud.gov/sites/dfiles/OCHCO/documents/CybersecurityIncidentResponsePlan2.0.pdf

[23] A. Applebaum, S. Johnson, M. Limiero, and M. Smith, "Playbook oriented cyber response," in *2018 National Cyber Summit (NCS)*. IEEE, 2018, pp. 8–15.

[24] G. Apruzzese, M. Andreolini, M. Colajanni, and M. Marchetti, "Hardening random forest cyber detectors against adversarial attacks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 427–439, 2020.

[25] J. Martínez Torres, C. Iglesias Comesaña, and P. J. García-Nieto, "Machine learning techniques applied to cybersecurity," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 10, pp. 2823–2836, 2019.

[26] A. Yeboah-Ofori, S. Islam, S. W. Lee, Z. U. Shamszaman, K. Muhammad, M. Altaf, and M. S. Al-Rakhami, "Cyber threat predictive analytics for improving cyber supply chain security," *IEEE Access*, vol. 9, pp. 94 318–94 337, 2021.

[27] S. Kalyani and K. Shanti Swarup, "Classification and assessment of power system security using multiclass svm," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 5, pp. 753–758, 2011.

[28] S. S. Sekharan and K. Kandasamy, "Profiling siem tools and correlation engines for security analytics," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 717–721.

[29] T. Kim, H. Lim, and J. Nah, "Analysis on fraud detection for internet service," *International Journal of Security and Its Applications*, vol. 7, pp. 275–284, 11 2013.

[30] C. Islam, M. A. Babar, and S. Nepal, "A multi-vocal review of security orchestration," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–45, 2019.

[31] J. Blankenship, C. O'Malley, S. Balaouras, A. Bouffard, and P. Dostie, "Forrester reprint," 12 2020. [Online]. Available: https://reprints2.forrester.com/#/assets/2/108/RES157496/report

[32] Splunk, “Enterprise security solutions,” Splunk, [Accessed Jan. 9, 2022]. [Online]. Available: https://www.splunk.com/en_us/software/enterprise-security.html

[33] Splunk, “Splunk software — phantom,” Splunk, 2020, [Accessed Jan. 9, 2022]. [Online]. Available: https://www.splunk.com/en_us/software/splunk-security-orchestration-and-automation.html

[34] G. Settanni, F. Skopik, Y. Shovgenya, R. Fiedler, M. Carolan, D. Conroy, K. Boettinger, M. Gall, G. Brost, C. Ponchel *et al.*, “A collaborative cyber incident management system for european interconnected critical infrastructures,” *Journal of Information Security and Applications*, vol. 34, pp. 166–182, 2017.

[35] IBM, “Enterprise security,” [Accessed Jan. 28, 2022]. [Online]. Available: https://www.ibm.com/security

[36] Microsoft, “Azure sentinel – cloud-native siem solution — microsoft azure,” azure.microsoft.com, [Accessed Jan. 30, 2022]. [Online]. Available: https://azure.microsoft.com/en-us/services/microsoft-sentinel/

[37] T. H. Project, “Thehive project,” www.thehive-project.org, [Accessed Jan. 30, 2022]. [Online]. Available: https://thehive-project.org/

[38] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, “On the effectiveness of machine and deep learning for cyber security,” in *2018 10th international conference on cyber Conflict (CyCon)*. IEEE, 2018, pp. 371–390.

[39] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, “Machine learning and deep learning methods for cybersecurity,” *IEEE Access*, vol. 6, pp. 35 365–35 381, 2018.

[40] Ł. Podlodowski and M. Kozłowski, “Application of xgboost to the cyber-security problem of detecting suspicious network traffic events,” in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 5902–5907.

[41] K. Ghanem, F. J. Aparicio-Navarro, K. G. Kyriakopoulos, S. Lambotharan, and J. A. Chambers, “Support vector machine for network intrusion and cyber-attack detection,” in *2017 Sensor Signal Processing for Defence Conference (SSPD)*, 2017, pp. 1–5.

[42] M. A. Siddiqui, J. W. Stokes, C. Seifert, E. Argyle, R. McCann, J. Neil, and J. Carroll, “Detecting cyber attacks using anomaly detection with explanations and expert feedback,”

in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2872–2876.

[43] T. Dias, N. Oliveira, N. Sousa, I. Praça, and O. Sousa, "A hybrid approach for an interpretable and explainable intrusion detection system," in *International Conference on Intelligent Systems Design and Applications*. Springer, 2022, pp. 1035–1045.

[44] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[45] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2003, pp. 220–237.

[46] P. Shukla, "Ml-ids: A machine learning approach to detect wormhole attacks in internet of things," in *2017 Intelligent Systems Conference (IntelliSys)*. IEEE, 2017, pp. 234–240.

[47] J. Carneiro, N. Oliveira, N. Sousa, E. Maia, and I. Praça, "Machine learning for network-based intrusion detection systems: an analysis of the cidds-001 dataset," in *International Symposium on Distributed Computing and Artificial Intelligence*. Springer, 2021, pp. 148–158.

[48] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "Flow-based benchmark data sets for intrusion detection," in *Proceedings of the 16th European Conference on Cyber Warfare and Security. ACPI*, 2017, pp. 361–369.

[49] P. Sornsuwit and S. Jaiyen, "A new hybrid machine learning for cybersecurity threat detection based on adaptive boosting," *Applied Artificial Intelligence*, vol. 33, no. 5, pp. 462–482, 2019.

[50] J. Sakhnini, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "Physical layer attack identification and localization in cyber–physical grid: An ensemble deep learning based approach," *Physical Communication*, vol. 47, p. 101394, 2021.

[51] J. B. Fraley and J. Cannady, "The promise of machine learning in cybersecurity," in *SoutheastCon 2017*, 2017, pp. 1–6.

[52] K. Highnam, K. Arulkumaran, Z. Hanif, and N. R. Jennings, "Beth dataset: real cybersecurity data for anomaly detection research," *TRAINING*, vol. 763, no. 66.88, p. 8, 2021.

[53] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, “Adversarial attacks on machine learning cybersecurity defences in industrial control systems,” *Journal of Information Security and Applications*, vol. 58, p. 102717, 2021.

[54] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, “Adversarial machine learning,” in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.

[55] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, “Addressing adversarial attacks against security systems based on machine learning,” in *2019 11th International Conference on Cyber Conflict (CyCon)*, vol. 900, 2019, pp. 1–18.

[56] J. Muniz, G. McIntyre, and N. AlFardan, *Security operations center: Building, operating, and maintaining your SOC*. Cisco Press, 2015.

[57] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[58] A. Mammone, M. Turchi, and N. Cristianini, “Support vector machines,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 3, pp. 283–289, 2009.

[59] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen *et al.*, “Xgboost: extreme gradient boosting,” *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[60] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[61] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, “Support vector method for novelty detection,” *Advances in neural information processing systems*, vol. 12, 1999.

[62] Z. Wang and X. Xue, “Multi-class support vector machine,” in *Support vector machines applications*. Springer, 2014, pp. 23–48.

[63] V. Franc and V. Hlavác, “Multi-class support vector machine,” in *2002 International Conference on Pattern Recognition*, vol. 2. IEEE, 2002, pp. 236–239.

[64] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “Knn model-based approach in classification,” in *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*. Springer, 2003, pp. 986–996.

[65] SATIE, “Security of air transport infrastructure of europe,” 2020, [Accessed June 28, 2022]. [Online]. Available: https://satie-h2020.eu/

[66] J. T. Hancock and T. M. Khoshgoftaar, “Survey on categorical data for neural networks,” *Journal of Big Data*, vol. 7, no. 1, pp. 1–41, 2020.

[67] K. Lange, “Singular value decomposition,” in *Numerical analysis for statisticians*. Springer, 2010, pp. 129–142.

[68] I. Bratko, “Machine learning: Between accuracy and interpretability,” in *Learning, networks and statistics*. Springer, 1997, pp. 163–177.

[69] C. W. Cleverdon, “On the inverse relationship of recall and precision,” *Journal of documentation*, 1972.

[70] C. Goutte and E. Gaussier, “A probabilistic interpretation of precision, recall and f-score, with implication for evaluation,” in *European conference on information retrieval*. Springer, 2005, pp. 345–359.

[71] S. Barnum, R. Martin, B. Worrell, and I. Kirillov, “The cybox language specification,” *The MITRE Corporation*, 2012.