UiO **Department of Informatics**
University of Oslo

A user-centric approach to explainable AI in a security operation center environment

A qualitative study on the use of SHAP and LIME to explain alarms

Håkon Svee Eriksson

30 Credit Master’s Thesis, Spring 2022

Abstract

Living in the information age, countries, societies, and individuals become ever more emerged in technology for each passing day. However, with every new software, hardware, protocol, and concept, comes a new possibility to manipulate and exploit it. Every year is a new global high in the number of cybercrime events, and the cost is expected to grow. Combine this with a slow and steady increase of unfilled cyber security positions and data encryption preventing detection, to create a problem in desperate need of attention.

The cyber security field will need to look for a solution, and perhaps a data driven subfield of artificial intelligence (AI), namely machine learning (ML) is one piece of technology in this big puzzle. With new records being set every year, the area of ML has shown itself useful in many industries and services. Cyber security have had an increased interest in utilizing ML for intrusion detection, given the huge amounts of data to analyze, and signatures of known malicious data proving less effective due to encryption. But a ML revolution have yet to reach the field. A part of the problem might be the model-centric approach, lacking a focus on the end-user. The analyst would in this case need to spend more of their precious time on understanding the alarm, rather than responding.

So how could existing technology be used to increase interpretability? This is where the field of explainable AI (XAI) is showing great promise. Even though the field's traditional end-user is a data scientist, this work bridges the gap to a security analyst. By utilizing two different methods from XAI called LIME and SHAP to generate intelligible alarms, security analysts could reach conclusions quicker, possibly stopping attacks before they even start.

This research present a human-centric approach when applying LIME and SHAP to an artificial neural network machine learning system for network alarm generation. This means that functionality is created with the security analyst's requirements in mind. The work in this thesis conducts an interview to identify characteristics of what an analyst believes a good alarm is, while discussing how the XAI methods meet those requirements.

The project have identified that good alarms make it clear what they are trying to detect, why that is important, while balancing between triggering both universal and precise, and being trustworthy by displaying the models certainty. Along with a specification of different alarm enrichment strategies, the research have shown that LIME and SHAP adds value to a security analyst, leading to interpretability and possibly a shorter analysis time. Furthermore, ways of making the new artificial neural network system co-exist with existing signature based solutions are discovered.

Acknowledgements

This master's thesis concludes the end of a two-year long master's degree in *information security* at the University of Oslo's department of informatics. I (the author) had already finished a bachelor's degree in *Programming and System Architecture* at the same department. The project is a part of the course *IN5930 Informatic. Master thesis* and have spanned from January to May in 2022. Adding up to a total of 17 weeks. The scope of this thesis is 30 credits.

The goal of the course is to both acquire an academic specialization in a field, as well as use that knowledge, combined with principles of responsible conduct of research to solve a problem. The study combines the field of explainable artificial intelligence and cyber security to generate good alarms for a security operations center. The amount of security personnel is not enough to cover the demand, and monitoring is an important aspect of security work. I was motivated to improve monitoring in the artificial intelligence field to help ease the shortage, while tackling encrypted malicious data.

I would like to give a huge thanks to principal scientist and colleague Gudmund Grov with the Norwegian Defence Research Establishment for his encouraging support, knowledge, and close scientific supervision. Additionally, a big thanks to professor Audun Jøsang, and Dr. Vasileios Mavroeidis with the University of Oslo for administrative supervision and positive feedback.

This thesis would also not have been possible, had it not been for every interviewee that prioritized time in their busy schedule, and shared their expertise and insight. The same applies to chief scientist Ann-Kristin Elstad, who provided guidance on how to conduct responsible interviews. A huge thanks to all of you.

Finally, I am extremely grateful for my family's support during a tough Covid-19 restricted and warmongering period. The same applies to friends and fellow students, with encouragement, bouldering, "guttastemning", and great cooperation. However, a special mention of a close friend that made life as a student richer. Victor, until we meet again.

Håkon Svee Eriksson
May, 2022
Oslo

Contents

List of Figures

Acronyms

XAI eXplainable Artificial Intelligence

AI Artificial Intelligence

SOC Security Operations Center

IoT Internet of Things

PPT People, Processes, and Technologies

ML Machine Learning

DARPA Defense Advanced Research Projects Agency

IOC Indicators of Compromise

IDS Intrusion Detection System

HIDS Host-based Intrusion Detection System

NIDS Network Intrusion Detection System

NIST The National Institute of Standards and Technology

DNS Domain Name System

C2 Command and Control

FP False Positive

CIA+P Confidentiality, Integrity, Availability and Privacy

ANN Artificial Neural Network

ReLU Rectified Linear Unit

Nmap Network Mapper

IPS Intrusion Prevention System

PDP Partial Dependence Plots

RNN Recurrent Neural Network

DNN Deep Neural Network

PyPI Python Package Index

NSD Norwegian Agency for Shared Services in Education and Research

FFI Norwegian Defence Research Establishment

TLP Traffic Light Protocol

CVE Common Vulnerabilities and Exposures

NAT Network Address Translation

POC Proof of Concept

CSA Cyber Situation Awareness

HAI Human-Automation Interaction

HCI Human Computer Interaction

TTP Tactics, Techniques, and Procedures

1 Introduction

1.1 Background and motivation

Cyber threats today are evolving at a speed many companies struggle to keep up with, and when it according to IBM, takes almost 200 days to just identify that a breach had happened [85], it shows that there is a significant portion of potential in bettering the detection aspect of an attack.

Traditionally, it is the Security Operations Center (SOC) that gathers logs from the company, analyzes it, and initiates a response. Alarms are commonly triggered based on a known malicious *signature*. But it is not sustainable to hire cyber security engineers to analyze huge amounts of encrypted data [99] (making payload inspecting signatures obsolete), leading to alarm fatigue [36, 198, 46]. So perhaps humanity should look for new data-driven solutions that, sees important unique feature combinations beyond encryption, detects previously unseen malware, and that can process a big amount of data fast.

Artificial Intelligence (AI) have recently grown popular because of its promising results in different areas spanning from medical to military defense. Its ability to process and analyze great amounts of data in a relatively short time makes it a great tool for the ever-increasing flow of data and information created from both the Internet of Things (IoT), and user behavior.

In this thesis, a focus is given to the data-driven part of AI, called machine learning (ML), to see how network alarm generation from an artificial neural network model, which is not considered interpretable (often referred to as a black-box) can be combined with two methods from the XAI field. Namely the local (single prediction focused) model-agnostic (can be used with all AI models) methods LIME and SHAP. Switching from providing explanation for a data scientist end user, to a security analyst. Alarm interpretability from black-box models are a known challenge in cyber security [178, 177], and the two methods should help address that, while increasing trust.

To judge if the new information generated from the two methods in the experiment are valuable, the research interviews SOC personnel. Defining characteristics for a good alarm, while gathering feedback from how the experiment have visualized the content to support intrusion detection. Especially the interaction part with a cyber analyst is lacking from surveys and research in the field [152, 87, 118, 143, 184]. The combination could complement a SOC with a new capability that eases the pressure of the cyber security profession shortage, while increasing model trust and managing to precisely utilize encrypted data.

1.2 Research questions

The goal of this thesis is to address the following research questions:

Q.1 What is considered as a good alarm in a SOC?

Q.2 Which requirements does a SOC have for alarms generated by machine learning?

Q.3 How can LIME and SHAP be used to create good alarms in a SOC?

1.3 Research method

In order to address the research questions, the following section will explain how the research was conducted, while clarifying some design choices. An overview is also given in figure 1. First of all, the project could be conducted as of a quantitative or statistical character. However, that would exclude important details explaining *why*, while possibly needing a larger time frame than this thesis, due to the data size requirement. Further detailing questions into specific concepts that might answer *why*, could prevent previously undiscovered factors from surfacing. Given the lack of previous research on analysts' interaction with XAI, one could say that this thesis puts more *weight* on exploration (qualitative), as opposed to exploitation (quantitative).

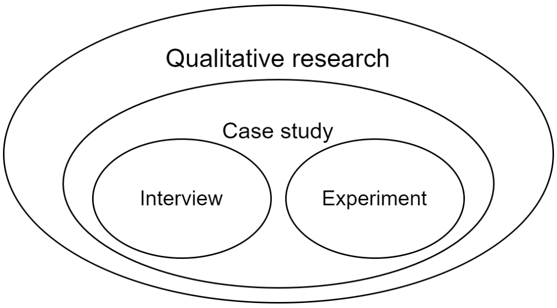


Figure 1: An overview of the relevant research methods in this thesis, and their domain.

Qualitative research has many possible approaches. Dr. Robert K. Yin, a renowned researcher in designing research studies, provides a good overview showing the different relevant situations of qualitative research approaches [200]. Based on (a) the form of research question, which is a *how* and *why* approach. (b) That the thesis does not need control over behavioral events, and (c) that the focus is on contemporary events (not historical). The *case study* approach, which aims to investigate real-world contemporary phenomenons in depth is recommended.

Following from a case study are data collection methods. Some common methods are survey, observation, secondary data, focus group, interview, and experiment [146, 145]. Given the length of this thesis, it is deemed realistic to only have time for two data collection methods. Combining more data points will help in data triangulation, further improving the case study validity. Since interaction with the two XAI methods LIME and SHAP is a central part, the project will first conduct an **experiment** to showcase the theory in practice. Creating and configuring not only a machine learning system, but also a signature-based for alarm comparison.

In choosing methods to validate the experiment, an important aspect is what kind of relevant previous research has been conducted. As mentioned in section 1.1, the field lacks an in-depth study of how the analyst might use XAI methods. Surveys are viable to gather opinions, however, given little research on this topic, there is an especially high risk of not covering important details in the questions. In comparison, focus groups and interviews would enable follow-up questions, possibly gaining a deeper understanding of the case.

Observation would be a good data collection method, especially when combined with an interview afterward. This combination would however need a tool the security analysts could use in the observation. Since this thesis's focus is the first steps on how such a tool may look (and if it is even relevant), it seems as if an observation study would be the natural next step as a part of future research.

Secondary data could also be a good addition to understanding what should be conducted when a given alarm appears. Yet there is a risk of it lacking information answering *why*. It is also highly doubtful that it will answer the questions regarding XAI, as that is a quite new topic regarding cyber security. The researcher would however see this method in combination with observation and/or interview as favorable.

Finally, considering that the project realistically would need to conduct the data collection during working hours for quick and easy access of personnel, and that most of the subjects would be affiliated to a SOC with an operative requirement, meaning that someone will always need to be on alert, monitoring and handling incidents. The most viable case given the time frame, suggested for the **interview** instead of a focus group. Seeing that one security analyst at a time can be pulled out from their work, it would minimally affect the operative team as a whole, while also being dynamic for rescheduling should a serious incident occur.

1.4 Contributions

The research conducted in this thesis uniquely contributes insight into how information from a machine learning system utilizing LIME and SHAP for alarm generation should be visualized for a security analyst (instead of a data scientist), and how it can co-exist with existing signature based intrusion detection systems.

The project also identifies a list of characteristics linked to a *good alarm* for a SOC, while comparing it to what LIME and SHAP can produce. The characteristics try to answer *what* the alarm is triggering on, and *why* that is important. Along with providing information to increase alarm trust, and finding the appropriate alarm priority.

Lastly the project, as part of the experimentation, contributes implementation of one ML model combined with two XAI methods (LIME and SHAP). The integration generates unique data used to better an analyst's investigation, which is supported by the conducted interview. The experiment and interview combination also reflects the thesis's contribution by bridging the cyber security fields' technical (alarm generation) and social operational parts (human alarm analysis).

1.5 Outline of the thesis

The rest of the thesis consists of 5 chapters:

2 Theory and literature review

This chapter will introduce some central theory and terminology that is important for the rest of the thesis, while taking accounts for existing literature. Concepts like AI, XAI and SOC regarding cyber security is described.

3 Experiment

Explains how the ML algorithm used for alarm generation is created, the XAI methods have been integrated, and the predictions designed for showcasing to an analyst developed.

4 Interview

The chapter regarding the interview uses the output from the experiment to ask participants questions related to the research questions, while detailing how the interview was conducted, and who participated.

5 Results and Discussion

Here the results from the research are presented and discussed.

6 Conclusion

The thesis is concluded with an assessment regarding the research questions, before ending with some proposals to future work.

2 Theory and literature review

In order to answer the research questions, some fundamental theories should be addressed. The work touches on multiple large categories like visual analytics and tool effectiveness for security analysts, studies on SOC operations and processes, and the use of machine learning in a cyber security domain. The active research fields contain a significant portion of theory, so in order to structure this part, the chapter will start by what a Security Operations Center and its alarms are, whit a scope and focus on network alarms. Secondly, it will introduce Artificial intelligence, and Explainable Artificial Intelligence. Lastly, the chapter will elaborate more on how the two areas can combine, looking at examples from related work.

2.1 Security Operation Center

A Security Operation Center is the unit in an organization that handles most of the security operations. The general goal usually evolves around strengthening the organization's security posture. Central functions include, but are not limited to cybersecurity threat detection, analysis, and response [192].

2.1.1 Teams

The unit is usually a combination of both operative and supporting teams. In the core are the level 1 analysts that constantly monitor real-time data, and thereby the alarms the data generates, while making initial analysis of incidents. After an alarm is deemed suspicious by a level 1 analyst, a level 2 analyst with more experience may take over the case for a closer inspection, before closing or escalating the alarm to an incident management team [76].

Depending on the company, the incident management team could be composed of both technical and non-technical entities. For the non-technical part, a manager or executive responsible for risk tolerance (Determining issues like if a production server should be shut down, or continue to run while possibly being compromised), legal and financial counsel, as well as a team leader determining the investigative steps might be present. Technical entities would include forensic investigators, IT and security staff.

Other supporting teams involve the engineers that setup and maintain the infrastructure. An intelligence team gathering knowledge on threats and vulnerabilities. A threat hunting team that proactively looks for possible non-alerted threats, and an offensive/red team that attacks the companies services in a controlled manner to unveil weaknesses.

2.1.2 Technology and logs

Tools

The SOC may have a wide range of tools in their disposal, some important once are a security information and event manager (SIEM), that collects logs and events from an Intrusion Detection System (IDS), such that an analyst can interact with it, and launch search queries [124]. Additionally, an email and/or ticketing system to document and escalate events, while communicating with other teams and company members is central.

Infrastructure

All of this technology will be organized by the engineering team in an infrastructure. The following is a brief introduction to some central components in an intrusion detection and prevention system:

- **Sensors**, monitoring the activity on a host or in a network.
- **Sensor administrator**, to make sensor management simpler and automated.
- **Database**, to store the reported activity from sensors and previous incidents.
- **Analysis tools**, can be standalone software, or a package of solutions that assist and even automates parts of the assessment process. Can correlate indicators across many sensors.
- **Interface**, that gathers information from different analysis tools, and presents the activity in a format easily understood by humans. The main tool security operation analysts interact with for alarm assessment.

Network intrusion detection logs

Many logs collected by the sensor might be interesting regarding network activity, they are *Netflow*, Packet Capture (PCAP), Firewall, Proxy, Browser history, and DNS to name a few. However, the project will mainly focus on Netflow and PCAP, since they are the ones generally present in publicly available datasets in the cyber security field.

The research will often refer to the concept of *flow*, which is short for *netflow*. Even though Cisco was the first to introduce the term as part of a function in their routers [81], it is commonly used to describe a combination of data regarding network protocols. So when data is sent over the internet, metrics defining what type of transport layer protocol is used, when it was sent, which IP addresses are transmitting, and the size is usually a part of Netflow. To differentiate it from Netflow which only collects a portion of the data and metrics being transmitted, PCAP data contains the full information (including payload). Netflow is also commonly represented as tabular data, which is data organized in a table of rows and columns.

2.2 Intrusion investigation

The following sections will focus on the processes in a SOC, by presenting some basic principles in intrusion investigation. Touching on aspects in incident response that includes detection, analysis, logs, and alarm assessment. These topics are important for giving a better understanding of how security analysts process alarms, before evaluating how the use of XAI can support that.

2.2.1 Incident response lifecycle

Incident response can quickly become overwhelming if no standardized approach is followed to handle potentially complex cases. It is therefore advantageous to follow defined steps and processes. The National Institute of Standards and Technology (NIST) has defined some phases in the incident response process which will briefly be explained, before elaborating on the most relevant one, here namely *Detection & Analysis*. Figure 2, illustrates the life cycle [33].

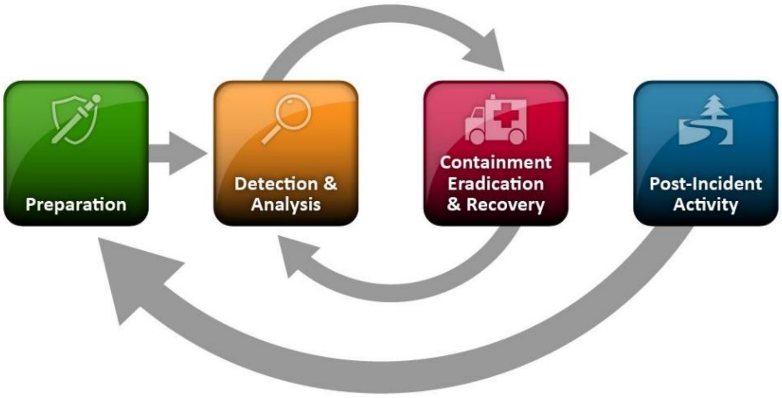


Figure 2: Incident Response Life Cycle [33]

Firstly, *preparation* resolves to create a response capability while securing systems and networks to prevent incidents. *Detection & analysis* involves instructions to handle common attack vectors, detecting an incident (relevant example from this thesis is via a Network Intrusion Detection System (NIDS)), gathering information from sources like network device logs, analyzing the incident, documenting the response before you finally can consider if the incident is worth notifying to another party.

Simultaneously with the analysis comes *Containment, Eradication & Recovery*. The step covers a containment strategy, evidence gathering in case of legal proceedings (An example from network is MAC and IP addresses), and then eradicating components that hinder the incident to continue (Deleting malware or shutting down compromised accounts) for then to initiate a clean recovery such that the system can operate as normal. Some recovery activities could include restoring backups, changing passwords, and even tuning firewall rules.

The last activity in the incident response life cycle is *Post-incident activity*. The team can then take some time to reflect on what happened, what they have learned, and what can be improved. An analyst should assess if the Indicators of Compromise (IOC)'s left after an incident can be made into signatures used in the IDS, which is a software that monitors devices and/or networks for malicious and violating activity.

2.2.2 Detection & Analysis process

When investigating a possible intrusion, the analyst will go through some steps in the detection and analysis process. The process includes the observed event, and interpreting it, before taking further steps to gather related data to make a complete analysis. Each step will be presented as a scientific method [93].

Observation

The method commonly starts with an **observation** or question. An observation can be split into two categories, namely *precursor* and *indicator*. The first is a sign of a possible future incident, while the other is a sign of an incident that has already happened, or is ongoing [33].

The analyst may have seen an event occurring in the IDS, which then kicks off the investigation. A precursor could be that the event noticed the use of a vulnerability scanner. An indicator scenario could be that a computer has done a Domain Name System (DNS) [1] query to a known malicious website, triggering an alarm. However it is worth noting that detection capabilities vary greatly, and can span from individual host-based intrusion detection systems to centralized log analyzers, or as this project is using, ML. To narrow it down, a general focus will be on network intrusion detection systems, or NIDS for short.

Hypothesis

Next, an **hypothesis** is built. The analyst will form an idea regarding the observation. From the example, the observer could theorize that a malicious program which sends out beaconing traffic to a Command and Control (C2) server [2] , is running on the computer. An important factor to mention when forming a hypothesis is the analyst's experience. A junior

[1] DNS is an internet service storing tables that tracks the connection of an IP-address (like 56.2.61.110) to an internet address (URL, like www.reuters.com)[159].

[2] An infected machine can be programmed to communicate with a server controlled by the attacker, to receive instructions and extract data to. The server sending the instructions is often called a C2 server [142].

analyst could form the given example, while a senior with previous knowledge of the sensor or signature could quickly assess the observation to an outcome in which the system incorrectly predicted the data as malicious, while it in fact was benevolent, also known as a False Positive (FP) [54].

Two additional topics will be considered as part of the hypothesis. Firstly if the observation can be *correlated* to other previously seen activity. If multiple different systems have reported on the same activity, it would strengthen the hypothesis, and the understanding of the incident. An analyst could look at time periods to combine events from multiple logs.

Secondly is ***event prioritization***. It is probable that an analyst has many observations to choose from, and must make an estimate of which event to conduct further investigation on [28]. Some relevant factors an analyst would consider are the effect on Confidentiality, Integrity, Availability and Privacy (CIA+P) [3] , its magnitude of impact, and recoverability.

Prediction

To continue the hypothesis, an analyst will make a **prediction** of what should be done, in order to find artifacts (tracks in data) that support the theory [194]. When referring to a possible malicious DNS query, evidence could be found in the query itself, network flows going to the answer of the query, and of course operating system, service, and application logs on the host in which the query originated [80]. Knowledge of the attack type and the sources available will quicken the process.

Other artifacts worth mentioning in a network focused case are the IP addresses, the port number of transport layer protocol (like TCP and UDP) headers . Specific string or hex values in the payload or header, and the number and size of packets (which could be abnormally high).

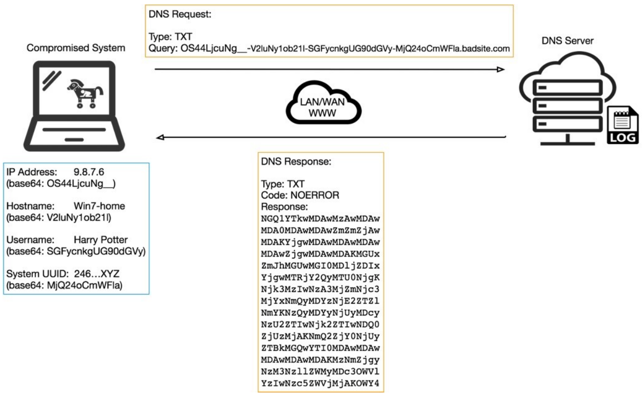
Experimentation and Testing

A prediction is made, and the next step in the method is **experimentation and testing**. The analyst will analyze the data to test if the hypothesis is right, by searching for artifacts. Using the example, artifacts showing DNS tunneling in the TXT field [4] of the DNS query as shown in figure 3, strengthens the hypothesis [149]. Another aspect of experimentation is to test for other potential explanations of the observed activity. Likewise, a correct hypothesis discards alternative theories (also known as falsification [12]).

[3] CIA+P represents four elements of security controls in information systems [166]. Confidentiality is to keep authorized restrictions on data [51]. Integrity is to protect against unauthorized altering of data, while ensuring non-repudiation and authenticity [52]. Availability guarantees that data is accessible when needed [50]. Privacy ensures that data regarding an entity is properly handled, focusing on collection, storing, managing, and sharing [53].

[4] A client can send different requests to a DNS server (Well known is an A record request, where the reply is the IPv4 address of the requested domain). One of which is a TXT lookup request, where the packets sent can contain strings of a size in which an acceptable bandwidth is achieved for a stable "tunneled" communication [149]

It is additionally some patterns in network traffic that could show signs of an incident. Two of those are: Systems that sends an unusual high amount of data in comparison to what they have incoming. Systems that utilize previously unseen ports and/or protocols, or a bigger range of them than normal.



```
DNS Request:

Type: TXT
Query: OS44LjcuNg__-V2luNy1ob21l-SGFycnkgUG90dGVy-MjQ24oCmWFla.badsite.com
```

```
IP Address:     9.8.7.6
(base64: OS44LjcuNg__)

Hostname:       Win7-home
(base64: V2luNy1ob21l)

Username:       Harry Potter
(base64: SGFycnkgUG90dGVy)

System UUID:  246...XYZ
(base64: MjQ24oCmWFla)
```

```
DNS Response:

Type: TXT
Code: NOERROR
Response:
NGQ1YTkwMDAwMzAwMDAw
MDA0MDAwMDAwZmZmZjAw
MDAKYjgwMDAwMDAwMDAw
MDAwZjgwMDAwMDAKMGUx
ZmJhMGUwMGI0MDljZDIx
YjgwMTRjY2QyMTU0NjgK
Njk3MzIwNzA3MjZmNjc3
MjYxNmQyMDYzNjE2ZTZl
NmYKNzQyMDYyNjUyMDcy
NzU2ZTIwNjk2ZTIwNDQ0
ZjUzMjAKNmQ2ZjY0NjUy
ZTBkMGQwYTI0MDAwMDAw
MDAwMDAwMDAKMzNmZjgy
NzM3NzllZWMyMDc3OWVl
YzIwNzc5ZWVjMjAKOWY4
```

Figure 3: Example of a C2 DNS tunneling in the TXT field [11].

Conclusion

Finally, the analyst will form a **conclusion** derived from the experimentation and testing. In the simple example, the artifacts of DNS tunneling supported the hypothesis. On the other hand, if DNS tunneling was not present, and the other log data did not support the hypothesis, the conclusion could state that it is falsified, or that more work must be conducted in order to reach a final conclusion. The work done should be documented to some degree, depending on company policy and procedures. They would also set the standard for evidence handling, communication, escalation procedures, and so forth.

From the detection and anlaysis process, multiple important investigative questions would have been answered. Some topics are:

- Longevity, of the malicious activity.
- Intention, behind the incident, if it was by accident or intended. By examining the characteristics of a virus, it might unveil the attackers intention (e.g. spying, destruction, financial gain ...).
- Additional, malware or malicious files.
- Scope, of the incident. How many systems have been confirmed effected, and what are the total potential of effected systems.

2.3 Monitoring

In order for a SOC to efficiently detect, analyze and respond, they have the process of monitoring. It gathers data from the company's network and servers. In the event that an anomalous activity occurs, the SOC will get an alarm [5] (usually generated based on past experience of known attack patterns) which they have to evaluate. Based on the evaluation, the necessary steps to investigate, defend and report can take place.

2.3.1 Signature and anomaly

There are two main methods used to detect anomalous activity, *signature-based* and *anomaly-based* detection. The former is the most used and was heavily relied upon by the earlier intrusion detection systems [10, 41]. It tries to create a unique *signature* for a known threat, such that it can easily be identified in the future. Examples of signatures are code patterns, the hash of a file, or an IP address. The content of these signatures are often referred to as IOCs [9].

A system based on clearly defined signatures can provide few false positives, and is easy to use and maintain. However, in order to construct a satisfactory signature, knowledge of the attack alongside information of the operating system version and application is key [100]. But as defenders improve, so do the attackers, with new specialized techniques like polymorphism, meant to avoid detection by regularly changing how the program appears while maintaining the same functionality [49].

In the case of NIDS (focusing on network data as opposed to a Host-based Intrusion Detection System (HIDS) which focuses on host data) where a typical IOC is an IP or a domain, the malicious actor can relatively easily modify their infrastructure to avoid future detection. However, it would take more time to transform their tools.

The *pain* given to the attacker when you detect their different indicators are commonly referred to as *The Pyramid of Pain* (Figure 4). In other words, how hard or easy is it for an adversary to change the indicator. At the bottom of the pyramid are hash values, both easy to detect and trivial to change with a small adjustment (e.g. polymorphism). At the top of the pyramid are Tactics, Techniques, and Procedures (TTP), which is both hard to change and detect.

So a question of how to climb the pyramid will naturally arise. This is where the other method of detecting anomalous activity becomes apparent. Because where signature-based are always reactive in nature (can

[5] Research might use the words alarm and alert interchangeably. This thesis will not put a significant portion of focus on the difference, but rather see an alert as a warning, of a possible future problem, while an alarm is considered a signal of a possible problem that must be dealt with (analyzed). In a cyber security setting, a group of alerts could raise an alarm. Similar definitions is found in [132], using *event* instead of alert, and alert instead of alarm.

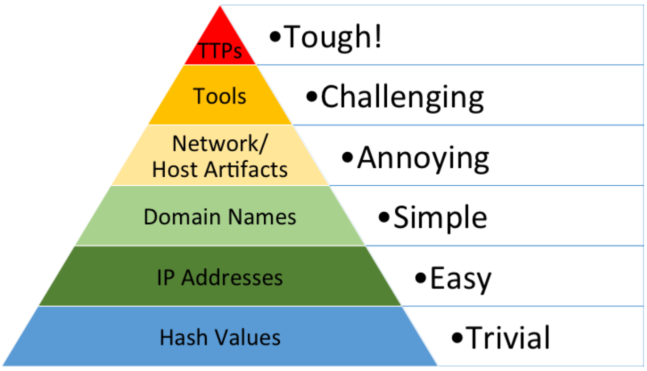


Figure 4: Model showing the relationship between an indicator to detect, and the amount of *pain* it will inflict the adversary if you are able to deny its usage [45] TTP denotes to *tactics, techniques and procedures*.

only identify known threats), *anomaly-based* detection will create a baseline of normal behavior, while continuously comparing new activity to the original. This gives the added benefit of discovering previously unknown threats, and even zero-days [6].

Since the anomaly-based method focuses on behavior rather than absolute values, it generally detects indicators higher on the pyramid of pain. However the rate of false positives is typically higher compared to *signature-based*, and whenever a company does a major change to the network (like launching new applications), the necessity of producing a new baseline might occur, leading to an increased demand for maintenance [63].

2.3.2 Encryption: The good, the bad, and the AI

As a transition to AI, it is common to mention big data (or in an analysts case, *big log data*), which might help reduce the challenge of data encryption. But first some more background information. Google can show reports since 2014 on the status of HTTPS adoption across the internet [67]. The data shows an almost doubling since then, growing from around 50% to 95% in 2022. Instinctively, network encryption improves online security by hindering eavesdropping from an attacker. However, the same mechanism is also widely used by malicious actors. A report from Zscaler found that "... 80% of attacks now use encrypted channels, up from 57% in last year's study" [202].

The problem becomes even more apparent when adding a survey conducted by SANS to the mix [41], where 45% of asked companies did not use any type of inspection on their encrypted communication. The same report also points out that 44% have inspection implemented, but that some

[6] A vulnerability that has not been disclosed to the public, and is often unknown to the vendor [25].

services are excluded because of “company policy and/or user privacy considerations”. So when most attacks are encrypted and the company does not provide inspection into the encrypted traffic, the ability to find patterns rise in importance. This is where ML unfolds as a probable candidate to tackle encryption for SOCs. Some projects have shown promise within this ongoing field [107, 16, 181, 13, 102], increasing confidence with the method.

XAIs relevance appears when inspecting an analyst’s process. Since there is a significant portion of different behavior features one could use (e.g. time, size, last 100 rows that share a specific characteristic...), the job to find which one (or combination) is central to the model’s assessment could be cumbersome. Pointing out relevant features might therefore be especially important when the payload is encrypted. Showing how explainability can bring projects tackling encryption one step closer to production.

2.4 Artificial intelligence and machine learning

Moving from intrusion investigation and monitoring, to a field impacting it more and more, the use of AI in anomaly based detection has been current for many years, and is still an area frequently researched [98]. The project will therefore present some theory on AI, and XAI, before further elaborating on its usage in a SOC environment.

Britannica’s definition of AI, is “The ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings” [37]. It can be seen as a system that processes data or observes an environment, for then to take any action that leads to a predefined goal. Now if humans depict the actions as intelligent, the system might get the term AI appointed to it. Surely this definition is quite broad, and this research will look into different sub-categories in order to better specify a possible operational usage.

The technology has gained popularity in a wide range of applications like speech recognition [17], search engines [199], and games [173]. Projects in cyber security span from malware monitoring [139], intrusion detection [185], alarm aggregation [106] and detection rule generation [195]. However, before taking a closer look at previous examples in cyber security, some more concepts in AI will de presented.

To further *narrow* the scope of this thesis, the research will be conducted in a part of AI called ML. ML focuses on learning/improving based on data, to recognize patterns without being specifically programmed. Typically with the use of historical data, future predictions can be made. Examples in this sub-field of AI broad from manufacturing, financial modeling, and marketing [92]. The field also tries to solve a range of different prediction problems. Some are:

- **Regression:** Predict continuous numbers, like estimating the price of a stock.

- **Clustering:** Finding the most similar other sample, like music suggestion.
- **Sequence prediction:** Predict what comes next, like the autofill function on smartphones.
- **Binary classification:** Predicts categorical variables, whit an output formed as one of two classes [101].

ML have also played an increased role in the cyber security field. Recent years have shown advancements in topics like email spam filtering [44, 14, 48, 44], malware detection in both mobile [112] and non-mobile devices [162, 89], and phishing website detection [151, 163, 164]. However the most relevant case for this project is intrusion detection, where Artificial Neural Network (ANN) is prominent [5, 56, 144, 39].

2.4.1 Artificial neural network

ANNs is one of many subclasses for ML. It is a popular choice when wanting to separate different entities/inputs into classes, also known as classification [134] [2]. The model's design and functionality draw inspiration from the network of neurons inside a human brain. An ANN is built up of units called artificial neurons (as can be seen in figure 5), connected with each other by coefficients (weights) in layers. Each neuron will be structured as having a weighted input (w1), a transfer function, an activation function, and an output. They can also be considered as having one of two states, active or inactive. It is here the weighted sum of inputs that mandates the state, and the weights are the ones being adjusted in the models learning process [3].

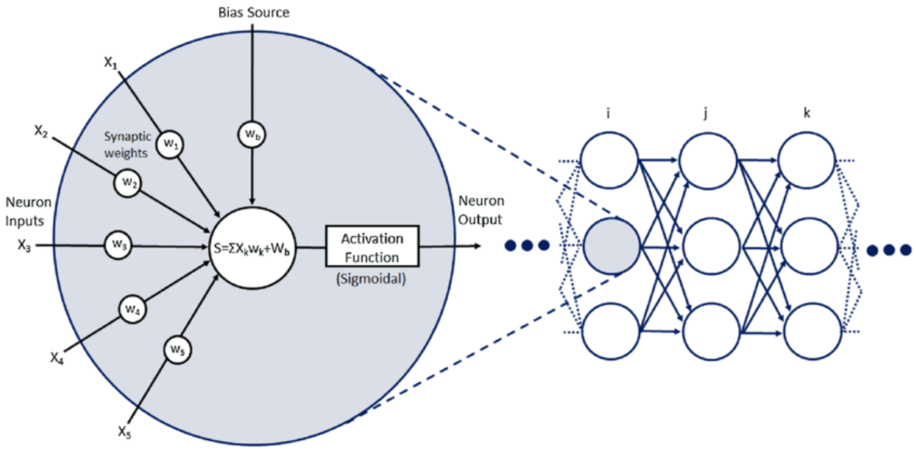


Figure 5: Depiction of how a typical ANN neuron functions [4].

2.4.2 Bias

ML algorithms depend on the quality and size of the training data. If that data is observations based on a preconceived notion of prejudice, the model is characterized as biased [79]. The consequence of bias in the case of cyber security, could lead to a reintroduction of intentionally excluded features [158], and be one factor to why models trained on openly available datasets have not been operationalized, as models rely on bias for their high accuracy [172]. Methods to mitigate bias focus on proportional representation of subjects in data, restricting adversary access to training data, and focus on maximizing average accuracy of each label (instead of one accuracy for all labels) [119, 179], to name a few. However to identify bias, the field of XAI have shown promise.

2.5 eXplainable Artificial Intelligence

An important aspect to describe is XAI. The topic for a common clear definition has been up for discussion in research, but since a prominent candidate has yet to reveal itself, this work will rely on the following definition: "Given a certain audience, explainability refers to the details and reasons a model gives to make its functioning clear or easy to understand." [21] As it explicitly reflects the explainability to a defined audience compared to other definitions [71]. In this project's case, explainability to a ML developer or researcher could be different than explainability to a security analyst.

To better express the importance of XAI, let's briefly look at the opposite. When it is hard to derive a good understanding of the model's inner workings after training (Like in the case of ANN's), the term *black box* is often used [31]. The challenge appears when these types of models are gaining popularity in systems that business crucial decision making are built upon. If the decisions can't be justifiable, legitimate or the actions of the system explained, a form of trust must develop with the model over time to know exactly when it is accurate and not. The consequence could be to shut down a part of the business, resulting in a huge cost. This type of trust would also take more time out of an already demanding workday for a SOC analyst, so the incentive to consider interpretability and explainability [7] when designing models is strong. As it helps to identify bias in the dataset and improve robustness [21].

The idea of trust is therefore quite central. The way XAI can build trust, is via different methods that try to shed light on why a decision has been made. The methods can focus on showing examples from the training data that is considered similar to the data predictions are made on, inputs (fea-

[7] Interpretability and explainability have some varying definitions, this thesis will be based on the following: "Interpretability is the degree to which a human can consistently predict the model's result"[97], and "Explainability is ... an interface between humans and a decision maker ... and comprehensible to humans" [21]. The definitions link interpretability as a passive model characteristic, while explainability is more of a process applied after prediction [122].

tures) from the data that was influential in the prediction, and explaining the model itself, as in what each layer of a neural network focuses on, and what features have been learned [22, 23, 160]. Other important terminologies and concepts are:

- **Causality:** The viewing of causality after you have trained the model, and before a decision is made for the test data [32]. In other words, can a human analyst derive a causal understanding from the model's output statement?
- **Transferability:** If you know how transferable a model is, you can make assumptions on whether or not a loss in accuracy on real-life data compared to the lab performance, is due to variabilities in the data from the two sets [105].
- **Informativeness:** A model could give more information to a security analyst than just the output. With the usage of an XAI method to enhance informativeness, the analyst can get extra information on what the model focused on in the input, before making a decision [65].

2.5.1 Intrinsic vs post hoc

Figure 6 shows an overview of classifies for the different methods in XAI. Starting from the top are two groups called *intrinsic methods* and *post hoc methods* [125]. In the first group, the model itself is considered interpretable. Examples of such models are decision trees [148], logistic regression [47], and linear regression [167]. The latter (post hoc) are methods applied after the training of a model is done, and as mentioned in one of Defense Advanced Research Projects Agency (DARPA)'s announcements on XAI, the highest performing models, are often the least explainable [20]. Models like ANN's are therefore not a part of the *intrinsic* group, and since a SOC environment demands a high accuracy on generated alarms [28], this work will focus on *post hoc methods*.

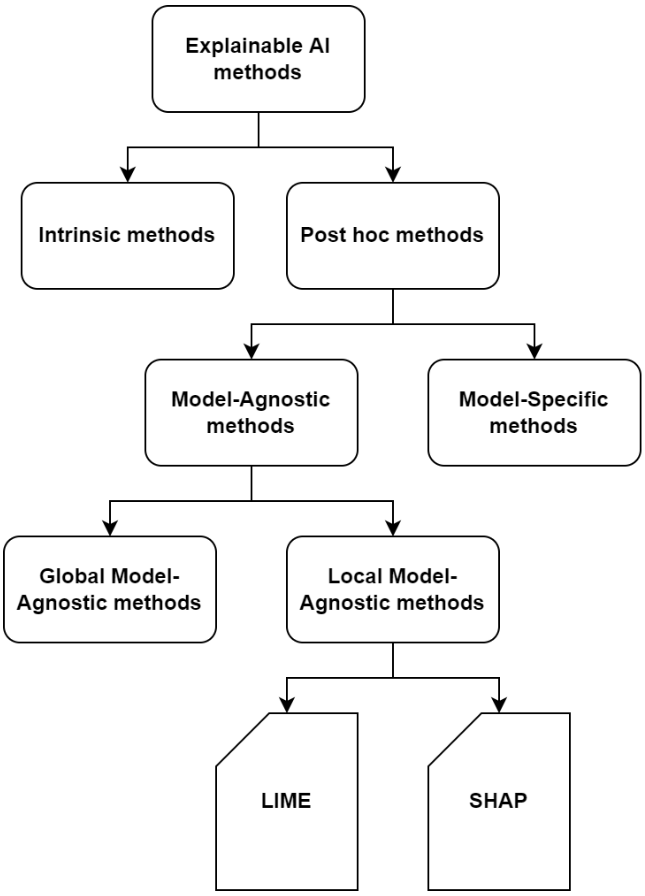


Figure 6: Overview of how XAI methods can be grouped, derived from theory in Molnar's book on interpretable machine learning [125]. LIME and SHAP are the focus of this thesis.

2.5.2 Model-agnostic vs model-specific methods

The *post hoc methods* can be divided into model-agnostic methods and model-specific methods. The main advantage as described by Ribeiro, Singh, and Guestrin [154], for model-agnostic methods, is their flexibility in being applied to any model, possibly making the process faster and easier for developers in the field. As well as flexibility in explanation to support both linear formulas and feature importance, and flexible representations of the data.

However, depending on the type of explanation that the end-user (or in our case the analyst) would like, model-specific methods could in some cases match that better. When looking at ANN that learn features from the hidden layers, a model-specific method could shed some light on what feature is learned, and which layer it is focusing on. When looking at the current discourse in the XAI field, most model-specific methods seemed to support image and text [165, 18, 133, 82, 138, 126, 59, 84]. The issue can also be viewed in Molnar's book on interpretable machine learning [125].

2.5.3 Global vs local model-agnostic methods

As shown in figure 6, model-agnostic methods can be further divided into global and local methods. The global method's perspective is the machine learning model as a whole, trying to showcase an average behavior. An example as described by Friedman is the Partial Dependence Plots (PDP) [91], that can display the marginal effect of a prediction for one or two features. In other words, the method shows the effect of a feature, when all others are marginalized. The PDP method does have a drawback since it assumes independence between features, which is often not the case, especially in cyber security. Another characterization of global methods is that some can explain which variable has had an important impact during training, and can therefore be convenient during development to adjust the model or the input data if issues such as bias appear.

Local model-agnostic methods on the other hand focus on explaining individual predictions. The focus shifts from feature importance for the model, to feature importance for a given prediction. The local methods aim to describe why the model made a certain classification for a given data point. In cases where the model is complex, a local view with some relatively simple rules could help describe an isolated event.

When looking back at the thesis's focus on a SOC, it is important to view the necessity of explanations from the analyst's view. A direct relation to local model-agnostic methods can be drawn as both the analyst and the method care about individual predictions (or a group of predictions). Operationally, the user of the model should not have to account for bias and other model weaknesses unless strictly necessary. So when asking "why does the model classify this data point as an attack?", local methods, compared to global methods should bring the analyst closer to an answer. Thus the focus is a local method henceforth.

2.5.4 Operationalized XAI methods

There are numerous methods in the local model-agnostic subclass. A 2021 paper survey discussed fourteen methods from only local to both global and local [29]. However, with implementation efficiency and the scope of the thesis in mind, two methods stood out as they were extensively used and referred to. Those were LIME [153], and SHAP [110]. They both have extensive tool support including Python libraries. This project will further on briefly explain how each of these works.

LIME - Local Interpretable Model-agnostic Explanations

LIME is a post-hoc method, supporting a local model-agnostic explanation. Given a model, the algorithm runs multiple times for the same prediction, altering different feature inputs to see how each feature affects the output. The altering is done based on a new dataset that the algorithm produces *around* the instance we are interested in, based on a real training dataset for the model (Figure 7). LIME is an intrinsic method, by creating an interpretable model, weighted based on how close each data point in the new dataset is to the instance we are trying to explain. The new interpretable model should give a good local prediction corresponding to the black-box model, and since it is interpretable, an explanation answering "why" is possible.

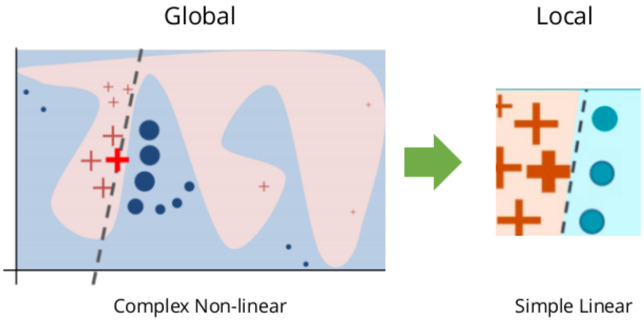


Figure 7: Example of how a global prediction compares to a local prediction of two classes (plus and circle). In the case of LIME, it generates new datapoints (Depicted as thin plus and circle) around the one it tries to explain (Depicted as thick plus). LIME can then create a linear interpretable model [116, 153].

SHAP - SHapley Additive exPlanations

The SHapley Additive exPlanations (or SHAP for short) is a ML explainability method by Lundberg and Lee [110] which like LIME, tries to explain individual predictions. SHAP values are actually based on Shapley values (Formulated by Shapley in 1952 [169]) from game theory, that in short tells us how to distribute a *payout* evenly among the players (or in our case, features) based on their contribution to a project/prediction. The value is generated from the average of all marginal contributions to every coalition. The marginal contribution is the difference between two predictions, where one has changed a feature with a random valid value (example shown in table 1). The number of marginal contributions will depend on the n number of features where 2^n is the total number of subsets being averaged for a single feature.

The goal for SHAP is to interpret a prediction by calculating how each feature contributed. The unique functionality with SHAP, contrary to only

Pred	**Source IP**	**Destination IP**	**S. Port**	**D. Port**	**...**
0,1	Random()	10.0.0.8	10	80	...
0,7	10.0.0.1	10.0.0.8	Random()	Random()	...
0,7	10.0.0.1	Random()	10	Random()	...
0,6	10.0.0.1	Random()	Random()	80	...
...	...	...	...	...	...

Table 1: Shows an example of how SHAP would create subsets of a datapoint. It *removes* values by substituting them with a random value from a representative dataset. *Pred* is the prediction score from the machine learning model. Note that without the value *source IP*, the prediction is low, indicating that it is influential.

Shapley values, is how the values are represented in an additive feature attribution method (linear model). Lundberg and Lee introduced different variations of the SHAP methods. This work will use an implementation similar to Kernel SHAP, called Deep SHAP[171]. The method is created for deep learning models and produces an approximate SHAP value. The method seeks to improve the computational performance by combining SHAP values from minor parts in the model with SHAP values for the model as a whole. This is done by feeding SHAP values backward through the network.

2.5.5 Advantages and disadvantages of LIME and SHAP

This section will present some advantages and disadvantages. Starting with the positive shared characteristics, both LIME and SHAP are model-independent. However, this project will for performance reasons use a SHAP implementation meant for deep neural networks. The method works for tabular, text, and image data, and is conveniently implemented in a Python library, making it easy to utilize in a Python codebase.

The Shapley values have a thorough theoretical foundation from game theory. This differs from LIME in how the prediction is fairly distributed among the feature values, since LIME does not guarantee a fair distribution. It is also worth noting that compared to Shapley values, SHAP is faster to compute.

LIME - Correct surroundings and robustness

Fast computation is not everything. The methods do come with some disadvantages. An unsolved problem regarding LIME with tabular data is its inability to give a correct definition of the surroundings regarding a data point. This consequently brings a need for tweaking kernel settings, to see if the method outcome makes sense. The Python functions used in this thesis support such setups.

LIME - Instability

Alvarez-Melis and Jaakkola [15] is focusing on **instabilities** regarding LIME's explanations. They show that relatively similar/close data points, could vary greatly regarding the explanation LIME produced. Further with a repeated sampling on the same data point, larger deviations occurred. This instability would isolate and make the LIME methods explanation harder to trust.

SHAP - Need of data and feature independence

The SHAP method also brings some challenges. SHAP, like Shapley, needs access to data to make a prediction. The prediction function alone is not enough, since the method randomly picks parts from the data to create new instances *around* the datapoint one wants an explanation of, to analyze the model's behavior.

SHAP and LIME - Picking non-representable data

Another aspect regarding the data is the assumption of feature independence both methods exercise when creating samples. LIME forms samples based on a Gaussian distribution, ignoring feature correlation. SHAP creates marginalized features by picking from the feature's marginal distribution. The procedure artificially generates a feature value that might not be present in the original dataset.

The practice of both methods performs works for independent features, although when considering data from cyber security, and especially network-based data which is heavily based on given protocols, the features are most likely dependent. This leads to a probable situation where samples not representable for the given domain are used to study a local explanation. The issue regarding SHAP is discussed in two papers [90, 183], presenting solutions that result in an output that no longer can be reflected as Shapley values.

SHAP and LIME - Hiding bias

Slack et al. [176] introduced another challenge. They managed to hide the biases of a black box classifier since LIME and SHAP are perturbation-based. Meaning that they modify the models input (like the port number of a flow datapoint), before observing how the output changes. Expecting important features to influence the output of a prediction [88]. Hiding bias results in methods creating misleading interpretations.

2.6 Explainability and the SOC

Explainability is important and could impact the SOC. The following section is a short presentation of related work on up-to-date uses, giving a view of how widespread it is.

2.6.1 Employment of explainability

A Gartner, Inc. report [64] predicted an 80% adoption of machine learning tools by 2024 , and a Micro Focus survey [120] claiming “over 93% of organizations use security operations products with ML or AI technology”, and mostly to detect advanced threats, indicates that a market for explainability is present. However the reports do not specify to what extent XAI are in use.

A 2018 study by Fernandes et al. [55], and an older 2014 study by Bhuyan, Bhattacharyya, and Kalita [24] both give a good overview of the different methods researchers have developed for intrusion detection systems. Yet none seem to even mention explainability as an issue, however interpretation is briefly mentioned when describing the role of a human analyst in network anomaly detection. Sommer and Paxson [178] already in 2010 identified the difficulty for an analyst to understand a machine learning model. The open issues mentioned in the papers, where XAI techniques could have an impact are the false alarm rates, reducing bias, and handling model complexity. A closer inspection of relevant literature finds a clear focus on accuracy [6, 57, 62]. In sum, the network anomaly detection field up to 2018 yields little focus to explainability.

After 2018 however, the XAI field combined with anomaly detection attracted some popularity. Wawrowski et al. [197] experimented with multiple classification methods for anomaly detection, while combining one of them (gradient boosting) with SHAP, calculating the coefficient for each variable. The work shows how XAI may be used to strengthen trust in the model, for example when brute force over Remote Desktop Protocol (RDP) [8] is conducted, the model sees the "RDP" feature as very important. This would naturally make sense for an analyst, and thereby possibly increase confidence in an alarm and the model on that type of behavior.

Karn et al. [95] implemented a ML for host-based anomaly detection in a Kubernetes cluster, with the goal of detecting crypto miners. To explain the model’s predictions, SHAP, LIME, and an autoencoder technique for Recurrent Neural Network (RNN) [9] models was used. The information from the XAI algorithms was used by developers and system administrators, to analyze the results, and build confidence in the model.

Mane and Rao [115] used a ANN to detect network intrusion, while presenting a XAI framework with the goal of improving the use of AI. It does so by focusing on interpretability in each step of the ML pipeline. Initially, interpretability and global post-hoc explanations are used such that a data scientist can correct, debug and improve the model. Then, when investigating a data point, the analyst is shown representative examples from the

[8] Remote Desktop Protocol provides a graphical user interface when connecting to another computer over the internet [121].

[9] Recurrent Neural networks are a neural network that focuses on a data’s sequential characteristics. Designed to handle temporal information (e.g. connected network packets, speech, or a sentence). The patterns derived are used to predict the next state. [2]

training data. Lastly, the end-user is given local post-hoc explanations to assist in understanding feature contributions.

Finally, Wang et al. [196] propose a framework with the intention of presenting an explanation for IDSs. They use SHAP's local capabilities to interpret single attack predictions, and the global functionality to highlight important features. This combination ties feature values and various attack types, while the earlier paper reserved global post-hoc explanations for a data scientist, here Wang et al. [196] have shown promise by promoting it as a form of enrichment, possibly gaining a deeper knowledge of attacks and their patterns.

To summarize, while most SOC solutions today use some sort of ML tools, research combining the explainable methods with cyber security seems to only have touched the surface as "proof of concepts" have emerged in the recent years. Operational practices accordingly suggest being in the minority.

3 Experiment

This chapter will explain a proof of concept experiment with XAI, where the output is used in the interview. The first part will explain how the two systems (signature-based and ML-based) are configured, before going into details of how LIME and SHAP are utilized. The project concludes that this will get the most realistic possible research materials for the interview.

3.1 Signature-based alarms

To generate signature-based alarms, *Suricata*, an open source signature-based detection engine [60] was used on a local machine. It is a widely used IDS for network activity, often compared to *Snort* [35, 72], which share many of the same functionalities. Suricata supports analysis and logging of a number of services like TLS/SSL, HTTP and DNS. Here, its PCAP processing capability is used, by generating a custom rule to detect a simple reconnaissance attack.

The attack

Signature-based IDS's are reliant on known attacks. The following paragraphs will go further into detail specifying the reconnaissance attack. To generate the attack, this research will utilize a popular tool called Network Mapper (Nmap) [70]. It is an open-source program that makes mapping out networks relatively fast and easy. With the help of raw IP packets, it can identify hosts on a network, their services (websites, ssh...), OS versions, and even details about the firewall.

```
$ nmap -sS 10.0.0.100
```

Listing 1: Nmap command utilizing the TCP SYN scan technique against IP 10.0.0.100

The option used to create a reconnaissance attack is the common TCP SYN scan as shown in Listing 1. The scan helps to detect a port state without establishing a full connection. A TCP/IP connection starts with a synchronization (SYN) packet being sent to the server, to initiate what is called a three-way handshake as described in RFC793 [86]. The server can now reply with an SYN/ACK, to acknowledge the synchronization as shown in Figure 8.

If the attacker is mapping a specific IP address, the program will note which ports are responding with a SYN/ACK, and mark them as *open*. The attacker will collect the information on open ports, and can now try to identify which service that is running on each of them, by sending more specific requests. It is also worth noting that continuously sending SYN packets to a server can potentially consume resources to a point where the activity becomes a *Denial of Service* (DOS) attack. Blocking for incoming legitimate traffic.

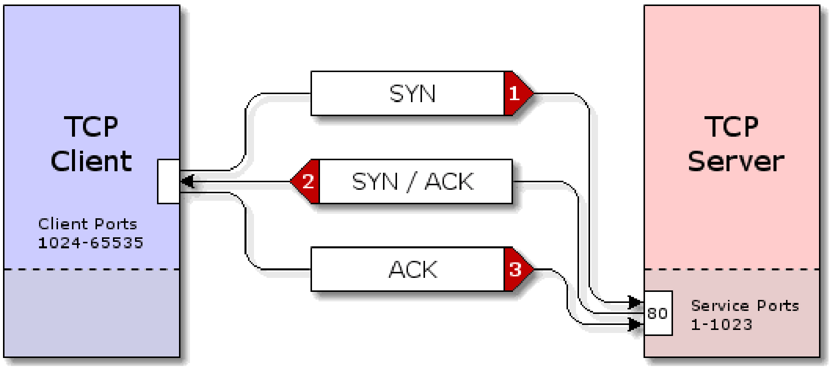


Figure 8: Visual model of a three-way handshake to a TCP server's port 80 [201].

The signature

Now that the attack is defined, a signature can be created to match the packet behavior. The details in the signature shown in Listing 2 will here be briefly explained based on Suricata's documentation version 6.0.4 [190]:

- **alert -** Tells Suricata to log the packet and generate an alert (which this projects for simplicity relates to an alarm). Suricata can also *drop* packets, and thereby function as an Intrusion Prevention System (IPS).
- **tcp -** Specifies the affected protocol. In this case, the Nmap attack is a TCP scan.
- **any any -** Means that the signature should look at packets where *any* IP address is going to and coming from *any* port. Smarter solutions like defining IP ranges and groups are possible.
- **-> -** Specifies which direction the packet of interest are going. In combination with *any any*, it is possible to match all packets, even the ones not answered, or just the answered ones.
- **msg -** The message that is shown to the analyst when the alarm is generated
- **flow:stateless -** Matches on packets that is both a part of an established connection and not. Flow can also be used to specify matches on flow direction (to the server only, to the client only ...)
- **flags:S,12 -** Indicates to look for packets where the SYN (S) flag is set while ignoring reserved bits 1 and 2, which according to RFC3168 [58] is the ECE (indicating if the TCP peer is capable to use Explicit Congestion functionality, which improves performance when a packet

drops) and CWR (Congestion Window Reduced flag used to by sender to communicate that it received a packet where the ECE flag was set) respectively.

- **classtype:attempted-recon -** Keyword used to classify rules and alarms. Defined in a config file with a short name, long name, and priority. In the config file for the presented field, attempted-recon would be the short name, while *Attempted Information Leak*, could be the long name. Priority will be shown with the alarm. Making it possible to prioritize different stages in the cyber kill chain. [10]

- **sid:2300000 -** sid is short for signature ID. Granting every signature their unique ID on a number format.

- **priority:10 -** Can be used to specify a priority ranging from 1 (high) to 255 (low), overriding classtype priority.

- **rev:1 -** Representing the signature version. For example, if the signature triggers a significant portion of false positives and undergoes a tuning, the *rev* number should increase, indicating a new version.

- **threshold:type threshold, track by_src, count 50, seconds 1 -** To determine if a reconnaissance attack is conducted, the amount of packets coming from one source is counted. The threshold type sets a minimum requirement before the alarm is generated. The requirement is 50 packets within 1 second from the same source IP. Additional requirements like flags will also need to be present in the packets.

```
alert tcp any any -> any any (
    msg:" Reconnaissance with nmap's SYN SCAN";
    flow:stateless;             flags:S,12;
    classtype:attempted-recon;  sid:2300000;
    priority:10;                rev:1;
    threshold:type threshold, track by_src,
              count 50, seconds1;)
```

Listing 2: The suricata signature used to detect NMAP TCP SYN scan

After creating a signature, it is placed in a *.rules* file, normally located in `/var/lib/suricata/rules`. However, for the system utilized in this work, Suricata with all the relevant files was stored in `/etc/suricata/`. Details of the config (`/etc/suricata/suricata.yaml`) will not be specified, other than to notice where the default logging directory is located, which for this project was `/var/log/suricata/`, and to include the newly generated signature file under the *rule-files:* option. The following line was created: `-/etc/suricata/rules/forskning.rules`

[10] The cyber kill chain is a famous model presented by Lockheed Martin in 2011, illustrating the series of steps conducted in a cyberattack [161]

Generating an alarm

Suricata can now be restarted, and a presentation of the logging conducted with the following commands:

```
  $ sudo suricata -c suricata.yaml -i ens18
  $ sudo tail -f /var/log/suricata/fast.log
```

The system is now ready to investigate network packets and alarm on reconnaissance in the form of a TCP SYN scanning. This work used another system on the same local network, to conduct the attack. Applying the command specified in Listing 1. The result was a raw alarm in the format: TIME - SID - NAME - CLASSIFICATION - PRIORITY - PROTOCOL - IP:PORT -> IP:PORT. An example is shown below, where IP ending in 91 is the system running Suricata, and the IP ending in 66 is conducting the Nmap scanning:

```
03/09/2022-20:37:49.833584  [**] [1:2300000:3]
Reconnaissance with nmap's SYN SCAN [**]
[Classification: Attempted Information Leak] [Priority: 10]
{TCP} 10.0.0.66:60522 -> 10.0.0.91:1309
```

By showcasing the raw data of a generated alarm, the signature-based system setup are concluded. The realistic alarm can now be used along with the signature itself in an interview.

3.2 Machine learning system

Next are an explanation of the machine learning system. Starting with a look at the different datasets commonly used in cyber security, and how the dataset has been altered (feature engineered) to fit as model input, before focusing on how the model is put together and trained. The section ends with a description of how the two XAI methods LIME and SHAP are implemented.

3.2.1 The dataset

When collecting data for NIDS, the traffic is usually collecting in either a *packet* or *flow* format. Packets can be captured by mirroring a network device's port, and, therefore, grasps the complete network information, with payload. Flow on the other hand does not contain a payload but consists of metadata. A common definition of network flow is the *five-tuple* variant as described in RFC6146 which includes the "source IP address, source port, destination IP address, destination port and transport protocol" [111]. However, when taking a closer look at the cyber security field's datasets, it is common to see additional attributes like *number of transmitted bytes*, *TCP flags* and *date*. Since the focus of this thesis is on alarm generation, and not feature extraction and model validation, a concept based on a dynamic (not strict to only five tuples) flow format is pursued.

The NIDS domain was previously known for having a challenge with the lack of public and representative datasets [74]. The issue was even stated as one of the biggest challenges for anomaly-based intrusion detection by Sommer & Paxson in 2010 [178]. Thankfully since then, some datasets viable as *benchmark* status have been published. A survey on intrusion detection datasets conducted in 2019 [156] gives a general recommendation of the following: CICIDS 2017 [170], CIDDS-001 [157], UGR'16 [113] and UNSW-NB15 [131] [129].

So which datasets might fit this research? Assuming the SOC have access to packets, they can extract metadata forming flows. Extensive use of metadata would be interesting in combination with a Deep Neural Network (DNN) model that draws context from huge datasets. The same would apply for XAI methods, giving them more features to examine. Secondly, to make the data driven alarm generation similar and comparable to the signature-based, leaving fewer different factors in the experiment, every ML model prediction should be based on individual flows (and not a batch). This demands some kind of connection in the features whit the previous flow.

The UNSW-NB15 dataset filled the first requirement by providing packets, but also the second with a finished flow extraction set, containing features that look at the previous 100 flows, and count how many of the rows that share service, source IP, destination IP, and more. A complete overview can be seen in appendix B.7. Since this kind of work was already done with the UNSW-NB15 dataset, this project have trained the model using that. It was also not so huge in size (45 MB, approximating to 258 000 lines of flow), making analysis, training, validation, and testing faster without special demand for hardware.

3.2.2 Feature engineering

The UNSW-NB15 dataset is released with pre-extracted flow data, containing not only the most common features like protocol usage and size of packets but also TCP sequence numbers, round-trip times, and specific features on HTTP traffic. These extra features add up to a total of 49 (50 with row index). Moustafa and Slay have a more in-depth view of each feature [130]. This work will not evaluate each one, but rather for simplicity do smaller changes that are explained in more detail below.

Isolate attacks

The first step in the feature engineering phase is to remove all other attacks, except *Reconnaissance*, which is in focus. Compared to the others, reconnaissance is relatively easy to simulate and generate signatures for, in order to create an alarm, thus making the comparison easier. Before more advanced functionality is added, three columns are removed: *id*, *attack_cat* and *label*. Representing a simple row index, name of the attack (ex. Reconnaissance), and a numeric value of 0 representing normal, or 1 representing an attack, respectively. The label column will however be used later in the training and validation phase. The process results in 47 total features, which is shown below. The table was assembled from different sources [75, 130].

Overview of all the features provided from UNSW-NB15 split into 6 groups

- **Flow data**: The identifyingfocused attributes between entities
- **Basic data**: Attributes which represent connection protocols
- **Content data**: Attributes for TCP/IP, and some HTTP services
- **Time data**: Attributes related to time, like time intervals between packets, start/stop, and RTT for TCP
- **Extra data**: Attributes meant to protect the protocols service, and relating one flow to the 100 previous ones
- **Attack data**: Attributes specifying if the flow is attack, and which type of attack

#	Name	Description	#	Name	Description
	FLOW DATA			**TIME DATA**	
1	Srcip	Source IP address	27	sjit	Source jitter.
2	Sport	Source port number	28	djit	Destination jitter.
3	Dstip	Destinations IP address	29	stime	Row start time.
4	Dsport	Destination port number	30	ltime	Row last time.
5	Proto	Protocol type (TCP, UDP…)	31	sintpkt	Source inter-packet arrival time packet arrival time.
	BASIC DATA		32	dintpkt	Destination inter packet arrival time.
6	State	The states and its dependent protocol e.g., CON.	33	tcprtt	Setup round trip time, the sum of 'synack' and 'ackdat'.
7	Dur	Row total duration.	34	synack	The time between the SYN and the SYN_ACK packets.
8	sbytes	Source to destination bytes.	35	ackdat	The time between the SYN_ACK and the ACK packets.
9	dbytes	Destination to source bytes.	36	is_sm_ips_ports	If srcip (1) = dstip (3) and sport (2) = dsport (4), assign 1 else 0.
10	Sttl	Source to destination time to live.		**EXTRA DATA**	
11	dttl	Destination to source time to live.	37	ct_state_ttl	No. of each state (6) according to values of sttl (10) and dttl (11).
12	sloss	Source packets retransmitted or dropped.	38	ct_flw_http_mthd	No. of methods such as Get and Post in http service.
13	dloss	Destination packets retransmitted or dropped.	39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
14	service	Such as http, ftp, smtp, ssh, dns and ftp data.	40	ct_ftp_cmd	No of flows that has a command in ftp session.
15	sload	Source bits per second.	41	ct_srv_src	No. of rows of the same service (14) and srcip (1) in 100 rows.
16	dload	Destination bits per second.	42	ct_srv_dst	No. of rows of the same service (14) and dstip (3) in 100 rows.
17	spkts	Source to destination packet count.	43	ct_dst_ltm	No. of rows of the same dstip (3) in 100 rows.
18	dpkts	Destination to source packet count.	44	ct_src_ ltm	No. of rows of the srcip (1) in 100 rows.
	CONTENT DATA		45	ct_src_dport_ltm	No of rows of the same srcip (1) and the dsport (4) in 100 rows.
19	swin	Source TCP window advertisement value.	46	ct_dst_sport_ltm	No of rows of the same dstip (3) and the sport (2) in 100 rows.
20	dwin	Destination TCP window advertisement value.	47	ct_dst_src_ltm	No of rows of the same srcip (1) and the dstip (3) in 100 records.
21	Stcpb	Source TCP base sequence number.		**ATTACK DATA**	
22	dtcpb	Destination TCP base sequence number.	48	Attack_cat	The name of each attack category (e.g. reconnaissance, DOS …)
23	smeansz	Mean of the packet size transmitted by the srcip.	49	Label	0 for normal, 1 for attack
24	dmeansz	Mean of the packet size transmitted by the dstip.			
25	trans_depth	The connection of http request/response transaction.			
26	res_bdy_len	The content size of the data transferred from http.			

One-hot encoding

Secondly, the columns *proto*, *state* and *service* describe the protocol type in use (ex. TCP), the state of the dependent protocol (ex. FIN), and the service in use (ex. ssh) is one-hot encoded, and the original columns are deleted. Meaning that if column *proto* had two unique values like TCP and UDP, it would after a one-hot encoding have 2 new columns replacing the old. One with the column name *proto_tcp*, and another named *proto_udp*. If the original rows value in *proto* was TCP, a 1 value will be set in the column named *proto_tcp*, while a 0 is set in the *proto_udp*.

Normalize

Some models are more sensitive to huge feature value differences, so normalization is often used. Increasing the experiment's realism. The features are min-max normalized. Rescaling the data value to [0,1], which seem to be beneficial regarding accuracy [168]. However, the effect diminished when the model grew larger, or when sample sizes increased. Additionally, the function makes it harder to map the input to the output.

3.2.3 The model

The following section contains work conducted to develop the ML model. Many of the design choices are motivated by a combination of fast implementation, and a theoretical fundament of common and best practices.

The ML model was created utilizing the Keras open-source framework [96]. It's an easy to use ANNs interface for the Google-developed *TensorFlow* library [68] and a popular choice in both industry and research. With simplicity and fast experimentation in mind, Keras makes for a perfect tool when taking the thesis's focus on the model's output into account, rather than a model with a high score.

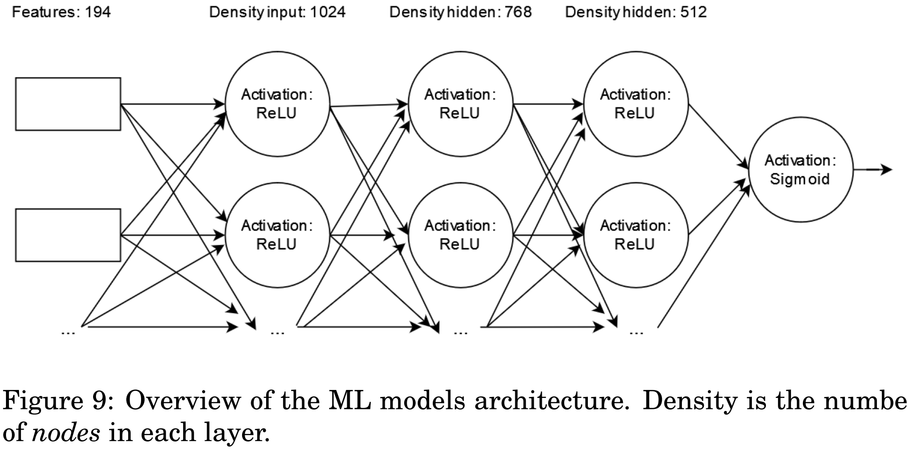


Figure 9: Overview of the ML models architecture. Density is the number of *nodes* in each layer.

The model (overview shown in figure 9) is based on Keras *Sequential* class. This class orders the layers of the ANN model into a linear stack, and therefore supports a single input and output sequence. Added onto the class are Keras's *Dense* layers [186]. These can be viewed as a regular connected ANN layer, with the dot product of the input and the weights matrix, added to the bias, before running into an activation function as seen in (1). This work utilizes the Rectified Linear Unit (ReLU) with the activation on each layer. In short, if the input is positive, that is output directly, otherwise, the default value is zero. ReLUs popularity can be justified by its ease of use in training and possibility of high-performance [77].

$$Dense\ layer\ output = ReLU(dot(input, weights\_matrix) + bias) \tag{1}$$

It should also be noted that in an operational setting, the model should support the analysis of continuous data streams (which is the case for network data). In addition, as a part of the investigation and threat hunting, it would be beneficial to inspect older data as well. The chosen model does support these features.

Input layer

With an introduction to how each layer functions, the project can take a closer look at the number of neurons in each. Starting with the input layer, the dimension is set to the number of features used from the UNSW-NB15 dataset. In this case that is the total number of features, minus the index, class, and label column. Index column are simply the index of the row, class is the name of the attack class (Analysis, Backdoor, DoS ...), and the label column is a numeric value of *0* indicating normal traffic and *1* for attack traffic. Ending with 194 in total features. Finally, the dimensionality of the output vector in this first layer is 1024. The choice of all output vectors was based on experiments with different sizes, and code from similar intrusion detection system projects, where Vigneswaran [193] is considered the most influential one.

Hidden and output layer

The two additional layers consist of 768 and 512 output vector units respectively. Between each of the total four layers is a *Dropout* class, that randomly sets the input units to 0 with a frequency of 0.01 to prevent overfitting [187]. Note that unaffected inputs are scaled by 1/(1-0.01), resulting in an unchanged sum. Finally, the output is mandated by the prediction problem the project is trying to solve.

This research is focusing on a binary **classification** problem where it should assign labels to choose whether a data point is part of an attack or just normal traffic. For this reason, the output layer is a single neuron, outputting a prediction between the interval of (0,1), thanks to the sigmoid activation function as defined in (2). Assigning the interval to a class is simply done with a split of 0.5. Everything greater is labeled as *attack*, while everything lesser is labeled *normal*.

$$Sigmoid(x) = \frac{e^x}{e^x + 1} \tag{2}$$

Number of layers and nodes

The number of hidden layers was chosen to represent a relatively easy architecture, while being fast to train. Since the goal is not a high accuracy, but rather explainability, from a model that in practice is unexplainable. It should be mentioned that Heaton [78] and Marsland's [117], both conclude that at most, only two hidden layers are necessary to "represent an arbitrary decision boundary to arbitrary accuracy ... and can approximate any smooth mapping to any accuracy.". Marsland also mentioned that it is possible to mathematically show that one hidden layer with many nodes is enough, and refers to this as *the Universal Approximation Theorem*. However the hidden layers size is not mentioned, other than that it is finite [83]. Possibly demanding exponentially more computational power than a deep neural network.

As portrayed in the research, the number of hidden layers seems to be thoroughly documented. The same can not be said for the number of hidden nodes/neurons in each layer. The same researchers mentioned that the prior view on the topic is to experiment with a different number of nodes and choose the one giving the best result according to your metric (often accuracy).

However, it is worth noting that too few nodes can result in underfitting, meaning there are not enough nodes to detect a clear context/signal in the data, while on the other hand, too many nodes may lead to overfitting. This could occur if there are more nodes to be trained than is data. It will also increase training time and energy consumption. Given that the thesis focuses on the cyber security field where huge amounts of data are not uncommon, a higher number of nodes might be advantageous. Nonetheless, as mentioned earlier, the focus is not to create a perfect model, but rather inspect the model's outcome. Minimal effort is therefore used for optimization and evaluation.

After the feature engineering and the model are constructed. Training is conducted with keras's *fit()* function [69]. The batch size is set to 64, and validation data is specified such that the loss is evaluated at the end of each epoch (one epoch is one round of passing the data through the model). Work done in [175], influenced this project for a similar setup, and the code for dataset splitting and training documented in [174] was lightly altered for this experiment.

The number of epochs ranged a bit, but a satisfactory accuracy was achieved with only 6 epochs, yielding little difference for each incremental increase. This can be seen from Table 2. Where *Loss* is the calculated

Epoch	**Loss**	**Acc**	**val_Loss**	**val_Acc**
1	0.1259	0.9533	0.0719	0.9735
2	0.0867	0.9699	0.1059	0.9614
3	0.0688	0.9771	0.0705	0.9763
4	0.0587	0.9803	0.0559	0.9777
5	0.0536	0.9819	0.0527	0.9796
6	0.0519	0.9822	0.0521	0.9826
7	0.0504	0.9834	0.0561	0.9801

Table 2: Table showing loss and accuracy for both the training set and test set (val) for each epoch.

distance between the prediction and the actual label (also known as ground truth). ANN adjusts its weights to minimize the loss value. The columns with *val_* are results from the test data, while the others are from the training data [27]. *val_Loss* can actually be used to prevent overfitting, a situation where the model fits the training data too well, resulting in worse real life predictions [26]. It is done by ending training when *val_Loss* stops decreasing, which in this case is 6.

3.2.4 LIME and SHAP

LIME and SHAP were both installed with the Python Package Index (PyPI) (Using commands shown in Listing 3), which is a repository of software for Python [147]. The implementation of both methods originates from two Github repositories. The LIME project is owned by Microsoft researcher Marco Tulio Ribeiro [155], and described in [153]. The SHAP project is owned by Microsoft researcher Scott Lundberg [109], and documented in [110].

```
$ pip install lime
$ pip install shap
```

Listing 3: Commands that were run to install LIME version 0.2.0.1, and SHAP version 0.40.0. Note that SHAP as of 02.05.2022 does have a newer version called “shap2” [108]. From the Github logs, it seems as if only smaller bug fixes and type checks have been conducted since pip's original shape.

LIME explainer

As of using the code from the repositories, LIME first creates what is called a *LIME explainer*. The explainer used in this project is the *lime_tabular* (since the data is of tabular format). Then function *explain_instance* is used to conduct a local analysis of the model's prediction. The explanations are saved to a `.html` file for later study. Listing 4 shows the complete function.

```
def xai_lime(model, train, test, columns, attack_indx):
    """attack_indx: Index of one attack flow in test dataset
    """

    # Create LIME explainer
    explainer = lime.lime_tabular.LimeTabularExplainer(
            train,
            feature_names=columns,
            class_names=["Normal", "Reconnaissance"],
            discretize_continuous=True)

    # Make explanation
    exp_attack = explainer.explain_instance(
            test[attack_indx],
            model.predict,
            num_features=len(columns),
            labels=(0,),
            top_labels=1)

    # Save prediction
    exp_attack.save_to_file("lime_prediction_attack.html")
```

Listing 4: This research's LIME function

SHAP explainer

SHAP is similar in the setup as compared to LIME. The method creates an explainer model called *DeepExplainer*. The explainer model is given a random set of 1000 rows from the training dataset, which it will use to learn how typical values in the different features occur. Next, the function *shap_values* is called on the explainer model to generate shap values for a given flow. The values are later sorted, and the top three features are printed. Listing 5 shows the most important part of the SHAP function.

```
def xai_shap(model, X, attack_indx):
    """ X: testdata
        attack_indx: index of one attack flow in testdata
    """

    # Create SHAP explainer
    r = np.random.choice(X.shape[0], 1000, replace=False)
    background = X[r]
    e = shap.DeepExplainer(model, background)

    # Create shap_values for one attack
    shap_values_attack = e.shap_values(X[attack_indx])
    shap_values_attack = shap_values_attack[0][0]
```

Listing 5: This research's SHAP function

Explaining the XAI

The system is now able to analyze which feature was locally important for a given prediction. However, after experimenting with how these values could have been shown and used by an analyst, an ambition to explain why a given feature was deemed important rose. This work therefore implemented a function that may be useful for an analyst when combining DNN's with XAI. The function is, an overview of how many of each prediction (normal and attack) have been seen for each unique value in a feature.

The function can be clarified with an example. An explainable model like SHAP is created and run on a prediction of the model, resulting in a set of SHAP values. The three features with the highest values (deemed as most influential) are put in a table, where each of their unique values is present. Lets say the three features from highest to lowest for a reconnaissance prediction is *proto_ipv6-no*, *sttl*, and *swin*. The unique values in the first feature are 0 and 1. In other words, the ipv6-no protocol is present in the network flow (1) or not (0). Listing 6 shows the first part of the functions output.

```
Feature             Score    Flow Value
-------------    --------   ------------
swin             0.113682              0
sttl             0.168806            254
proto_ipv6-no    0.189224              0
```

Listing 6: This research's function that tries to explain why a feature gained a high XAI method score, by *understanding regular traffic*. Showing the three features with the highest score of a reconnaissance attack datapoint. The score is SHAP (similar can be given from LIME), while *Flow Value* is the value the datapoint being predicted on had.

Now that the unique values are known, the training data can be analyzed by taking the summation of how many data points/rows are labeled *normal* with each unique value. The same is done for each labeled *attack* (which in this case is only reconnaissance). The result tries to give the analyst insight into how ordinary a feature value might be. Listing 7 depicts the generated table.

The following detail are important for understanding the table. The $*$ mark, located in the *swin* value table (Listing 7). The sequence 245...5* is to indicate every value as of 5, even 245. They are not displayed individually, since there are no attacks with these values. Note that the same type of concatenation could be used on the *sttl* value table.

The information from both of the listings (6 and 7) can now be transformed into a textual explanation for an alarm created by the ML-based DNN model, illustrated below:

The alarm triggered due to the following:

```
  swin value    Normal    Attack
------------  --------  --------
           0     26031      7022
        245...5*    20         0
         255     66949      6965

  sttl value    Normal    Attack
------------  --------  --------
           0      3846        24
           1       102         0
          29        52         0
          30         2         0
          31     56157         0
          32        19         0
          60        30         0
          62      6296        40
          63        32         0
          64       181         0
         252         2         0
         254     26279     13922
         255         2         1

  proto_ipv6-no value    Normal    Attack
---------------------  --------  --------
                    0     93000     13980
                    1         7         0
```

Listing 7: This research's function that tries to explain why a feature gained a high XAI method score, by *understanding regular traffic*. Showing the number of datapoints with each features unique value. Normal and attack represent the sum of datapoints with that label.

- Most importantly, the ipv6-no value was 0.
  - 99% of all normal traffic has that value, while 100% of all attacks have the same value.
- The sttl value was 254.
  - 28% of all normal traffic has that value, while 99,5% of all attacks have the same value.
- The swin value was 0.
  - 28% of all normal traffic has that value, while 50% of all attacks have the same value.

By creating a textual presentation of the data, the barrier of entry for using this kind of ML-based system could decrease. The system may bring a significant explanation to an alarm, such that the next steps in the analysis process are more targeted and faster, as compared to a system without

XAI. This concludes the experiment chapter, by creating two alarms of the same attack, from two implemented systems (signature and ML-based). The alarms, alongside the functions used in cooperation with LIME and SHAP, is now ready to be used as basis of interview questions.

4 Interview

After the experiment created the groundwork of how XAI can be shown and used, as well as providing realistic alarm data, an interview was conducted to deliver necessary feedback. This chapter explains how the interview was run, who the candidates were, how the data was analyzed, and a discussion of tactics used to increase the quantitative research's trustworthiness.

4.1 Conducting the interview

This work have followed the interview method described in Elstad's PhD thesis [19]. To ensure a good portion of exploration, the interview was done in a semi-structured way. Enabling follow-up questions on the fly if the answer was vague and probing to clarify concepts. The interview guide tried to reflect a balanced combination of open-ended questions and more specified ones, putting a little restriction on the candidate, while answering the thesis's research questions. An English translated version added to the appendix will be referred to progressively B. The original Norwegian version is also added to the appendix C.

4.2 Introduction

As part of the introduction, the interviewer presented their name, education, and work. Then a more general introduction to the purpose of the interview, was given. The candidates were informed of anonymity and the possibility of withdrawal from the study, in both plural and textual form (as part of the statement of the consent, see appendix D).

If consent for use of a recorder was given, the recorder would start just before the candidate was asked some introductory questions about themselves. To ensure anonymity, only high-level details was described. For example, when asked about education, only the degree level (high school, bachelor, or master's) and if they thought it was relevant were stated. Their subjective perception of experience in the field of cyber security, as well as AI, was graded with low, medium, or high. The introduction should contribute to making the candidate relaxed and comfortable.

4.3 Main part

To get a better understanding of how XAI supported alarms could be integrated with an analyst, the main part first tried to figure out if there is anything that can be classified as a *good* or *bad* alarm. If so, follow-up questions asking to explain their characteristics followed (Linking it directly to research question **Q.1**). On the occasion where candidates got stuck, goals

Time	SID	Alarm name	Classif-ication	Pri.	Prot.	From ->To IP:PORT
03/09/ 2022 - 20:37: 49.8	2300 000	Reconnaissance with nmap's SYN SCAN	Attempted Information Leak	10	TCP	10.0.0.99: 60522 -> 10.0.0.100: 1309
03/09/ 2022 - 20:37: 49.8	ML_ ANN_R	Possible reconnaissance activity	Attempted Information Leak	10	TCP	[3f9a:9e65: :e118:4f87]: 60522 -> 10.0.0.100: 1309

Table 3: Showing both alarm entries shown in the interview. The first is signature-based, while the second is ML-based. SID is unique identification. *Pri.* is short for priority. *Prot.* is short for protocol.

and attributes considered positive for an alarm like interpretability, correlation, and classification was actively used.

4.3.1 Alarm evaluation case

The candidate was then presented with a case (See appendix B.3), describing their role in a team of SOC analysts that have created both a signature-based and ML-based machine learning method for network detection (see appendix B.4). The attack type to detect was reconnaissance. The interviewee were then told that two alarms show up on their screen (Seen in table 3), where one is a signature based, and the other a ML based. Together with the interviewer, they went over each of the details of the alarm, such that the analyst was in a position to evaluate the methods.

In the case of the alarm originating from a signature, every value in the rule were explained. However, like the one derived from the ML model, a little more time was spent describing every part. Here the candidate needed to understand how the textual summary was formed, based on the XAI's top three features (Table 4) drawn from the complete flow-data (Appendix B.7 as well as the overview in 3.2.2, explains each feature), with information from the *explaining the XAI* function (see 3.2.4).

4.3.2 Case specific questions

The questions related to the case focused on making the interviewee describe how they felt the signature and ML methods differ. By identifying their perception, it might be easier to create functions supporting the analyst. This is also a central focus the thesis contributes to the field. Instead of targeting a high model accuracy, as what could be called a model-centric approach, this work prioritizes an analytical-centric approach. Placing the

SHAP:

Flow column	Flow value	Importance (SUM 8.2)	Description
proto_ipv6-no	Ipv6-no	0.18	No next header for IPv6
Sttl	254	0.14	Source to destination TTL
swin	0	0.09	Source TCP window advertisement value

Table 4: Displays how the three features/columns with the highest SHAP score was presented to the interviewee. *SUM* is the summation of SHAP values from all features. LIME had a similar table.

analysts need first. Good results on a model are one thing, however, the operational value is only present when it is put to use as an independent technology, or of what seems to be more often than not, in cooperation with an analyst.

Further questions focused on if the interviewee experienced that anything was missing, to make a good evaluation of the alarms. Follow-ups focused on the ML method, and especially the new functionalities like the textual summary. The last case-specific question tried to identify if the analysts had any thoughts on how the two methods could support each other. Detecting new synergy effects may increase the value output from the technology. Leading to more precise detection, lowering alarm fatigue [11].

4.3.3 General questions regarding anomaly

The project also took the opportunity to ask a few general questions regarding anomaly-based detection methods. Collecting the interviewee's thoughts on how an analyst with experience in only signature-based methods, better can understand that anomaly is simply seen as not normal, differentiated from a signature where the basis is hostile activity. Secondly, an *alarm per ip* function that tracks how many alarms a suspicious IP has generated would be helpful. The challenge and function were mentioned in earlier research and could reveal additional measures to ease the analysis.

4.4 Ending the interview

After the candidate has reflected on the possibilities and challenges of both the signature and ML method, they were asked to give a short evaluation

[11] Alarm/Alert fatigue is the state wherein this case a security analyst is exposed to a large number of alarms (Often false positives), desensitizing them to the threat. Increasing the chance that important alarms have an increased response time, or that it is not taken care of at all. [7, 40]

of low, medium, or high, on topics concerning both methods. The topics focus on interpretability, time used in the assessment, and the knowledge and intellectual demand it claims from an analyst. The questions sum up the interviewee's considerations, thereby providing a logical form of final remarks to the interview.

With the semi-structured interview coming to an end, it is considered good practice to summarize the session and clear up ambiguities [150]. The candidate is also encouraged to mention matters previously not discussed, that they consider relevant. Before concluding the interview with a final open question on what advice the analyst would give developers producing ML-based solutions for a SOC.

4.5 The candidates

The following section will explain who participated, and how they were recruited. A short discussion on data saturation and consent is also included.

To provide a better context of the data collected in the interview, some demographic questions were conducted. Given the complex quality of a cyber security role, it would be interesting to observe how different interviewees with unequal backgrounds perceived alarms, ML-based models, and XAI. However, on a general level, the research problem is not considered dependent on understanding certain demographic characteristics.

A focus on anonymity was a prerequisite for conducting the interviews. Also given the number of interviewees, which concluded to 10, details into every possible demographic feature (age, sex, detailed education information) were not possible. So to keep details at a minimum, only education, profession, work, and AI experience, along with involvement in network- and client-based alarms was included.

Recruitment and profession

Since the focus of the thesis in part is the usability of XAI methods as part of an ML detection system for a SOC, the natural main demographics should have a profession as a SOC analyst, or as a direct supporting role. That goal was achieved by recruiting from a SOC environment in Norway, as can be seen from the interviewee's answers to what they considered to be central work tasks. Their work spanned from analyzing technical network data and SOC infrastructure maintenance to more host-analysis and AI-based tool development. The self-reported experience in their work was on average high.

Education

Every candidate meant that their education was relevant to the job, whereas a majority held a master's degree. Most had a low level of previous experience with AI and ML, while four was evenly distributed among medium and high. Nearly everyone has had some sort of practice with analyzing and classifying both network- and host-based alarms, using a smaller portion (Median: 10%) of their active working time on the activity. To summarize, the interviewees were generally highly educated, with limited knowledge of AI, and considerable knowledge in alarm analysis.

4.5.1 Statement of consent

As part of the interview introduction, the interviewees were presented with an information sheet detailing the research's content, who is responsible, their rights, detailing privacy issues, as well as contact information if they want more information in the future. The statement of consent was actively checked and signed by every interviewee. The original Norwegian version is added as an appendix D. The interview has also been evaluated by the *Norwegian Agency for Shared Services in Education and Research (NSD)*, a national center and archive for research data. The center advised on data management and data protection.

4.5.2 Data saturation

The sampling done in qualitative research will be guided by the volume of data necessary to properly answer the research question. Fush and Ness [61] describe the stage in which new data from candidates adds less value than the energy and time needed as *saturation*. After each interview, the project evaluated the amount of data, that could be considered new and relevant. The project ended with 10 interviewees, before seeing that little to no new data was presented.

4.6 Data analysis

When the point of data saturation was reached, the analysis work started. The goal was to identify, categorize and trim raw data into conclusions [123]. Since some of the interviewee's interviews were recorded, the project started to transcribe these into text. After transcribing, the recordings were deleted according to the agreement specified in the *Statement of consent* document (Appendix D).

The textual answers were first organized in the themes of which the interview guide was structured. (1) Introducing the analyst, (2) Alarm definition, (3) Good alarm, (4) Case questions, (5) General machine learning, (6) statement evaluation, (7) Ending remarks. However, given the nature of the semi-structured approach, some of the topics were changed and the order in which the answers were given was shuffled, to present a better overview and understanding.

4.7 Quality and trustworthiness in qualitative research

A strategy used to increase quality was to contact a senior researcher at Norwegian Defence Research Establishment (FFI) with case study interview's as their field. The researcher's doctoral thesis assisted in finding research on case studies, as well as providing inspiration for the research design's structure and content [19].

To establish trust in qualitative research, tactics regarding validity, reliability, and researcher bias were implemented and discussed. The following section explains why and how in more detail.

4.7.1 Validity and reliability

Yin's, book on designing case studies [200], explained some criteria to evaluate research design quality. The criteria is presented as four *tests*, also found in other textbooks [94]. These are *construct validity*, *internal validity*, *external validity* and *reliability*. Following are how this work took some initiatives to improve these, by following the book's recommended tactics where it was seen as relevant.

Construct validity

Construct validity is selecting good measures for the studied concept. Since some of the interviews is focusing on identifying these measures (like in the case of identifying what a good alarm is), those could not bring construct validity.

Nonetheless towards the end, a grade of small, medium, and high was given to different statements. One concept mentioned in the statement is *assessment time*. The tactic described in Yin's book is to find other sources of evidence that in this case would compare the assessment time of signature- and ML-based detection systems. That was not found. A second tactic was having a key informant review the case study report draft. The informant, a manager at the SOC participating in the interview, did not challenge any of the reported findings. However, some results were elaborated on, by looking at the transcribed material, to clearly describe the interviewee's thoughts. This also reduced the likelihood of researcher bias.

Internal validity

Internal validit in this project concern the causal relationship between a good alarm leading to a shorter assessment time and less FP's. However since one of the research questions relates to exploring the characteristics of a good alarm, not seeking to form such casual relationships, this test is not relevant.

External validity

External validity tries to determine the generalisability of the findings from a study. To support this task, the research has briefed on the total number of candidates, along with their demographics. Every interviewee was also evaluated beforehand, to see if they were found appropriate to answer the research questions. The main criteria were their active connection to a SOC environment or known previous experience.

It is worth repeating that the interviewees are linked to the same SOC environment, weakening the generalisability of the observed data. Another point is the focus on network alarms presented during the case part of the interview. However the start of the interview asks for alarms with a general view, and as seen from the demographic, most interviewees had experience in both network- and host-based fields. So the thesis sees the generalisability of alarm as valid.

Additionally only one type of attack is considered (reconnaissance), and it is linked to one attack scenario (number of connections from the same IP, see Listing 2). There is almost no context other than the alarms given, and the XAI methods are of similar character (post-hoc, local model-agnostic). To increase generalisability, all of the mentioned aspects should be swapped, tested, and compared.

Reliability

Reliability holds the function of making sure that the findings in this work can be repeated, creating the same results. The research improves its reliability by thoroughly documenting how the case studies were conducted in the sections regarding experiments and interviews. Another tactic mentioned by Yin is a *case study database*, an orderly compilation of all data. Such an overview with all of the thesis's data was created, containing transcripts, sources (e.g. Yin's book), and even the different versions of the interview guide, along with version change notes.

4.7.2 Researcher bias

By shedding light on the investigator conducting the research's bias (also called *investigator's position*), possible discrepancies and prejudice may be accounted for when assessing the research [135, 180]. To address the researcher's bias, a personal statement is included.

At the time of writing, the author of this thesis was pursuing a master's degree in his last year at the University of Oslo, studying information security. The education is a combination of both technical (e.g. Ethical hacking) and governance-focused subjects (e.g. Security and Risk management). A bachelor's in informatics with a focus on software development was completed beforehand.

In parallel, the researcher worked with The Norwegian National Security Authority (NSM), The Norwegian Defence Research Establishment (FFI), and Sopra Steria. Gaining experience in both the cyber security and ML field. No family members have worked with cyber security.

The author's goal with this master thesis was to increase his understanding of processes in a SOC, how the field of AI can integrate with cyber security, and learn of the possibilities with XAI. The most surprising newly attained knowledge was all the details that go into designing and quality-checking qualitative research. Especially the interview, where recording demands an application to NSD.

Since the researcher had direct contact with the candidates, an awareness of how his personal, educational, and work experience might influence his data understanding and preparation was always present. Additionally, cyber security is not particularly known to be completely transparent. The field often utilizes the Traffic Light Protocol (TLP) to facilitate information sharing [12]. This might result in insufficient trust regarding how the author handles sensitive information, impacting the data. The balance of maintaining a social distance for the sake of the research, while building trust was challenging.

4.8 Ethical discussion

Before the interview, an ethical evaluation was conducted. The discussion is here presented in short. First of all, the use of ML might be seen as more unethical if the model is not explainable. However, the project would argue that the techniques used from XAI is making the use of ML more ethical, since it provides insight into the detection & analysis process, unveiling unwanted bias.

Another part of the ethical discussion would be to consider the consequences if the model created in the experiment generated a false alarm (FP) multiple times on the same IP because the role of the person is simply considered an anomaly as compared to the network as a whole. In the short term, this could lead to unintentional surveillance of individual people. On the contrary, if the model is operationalized, it should be given a feedback loop where the analyst would label the alarming data. So in the long term, the models would learn, thereby deviating from that behavior.

The ethical aspect of the interview is already short previously discussed by taking into account privacy assessments from NSD, operational security, as well as a content assessment from the leader of the SOC environment where the interview was conducted. The goal of the content assessment was to make sure that the results did not contain any wrong and sensitive information.

[12] The TLP is a set of usually four colors setting boundaries to whom the information may be shared with [34]

5 Results and Discussion

5.1 Results

In this section, key findings from the experiment and interview are presented. It is almost organized like the interview guide (see appendix B), with small changes as an adaption of the candidate's formulated answers. The subsections are closely related to the research questions, while its paragraphs focus on individual topics.

5.1.1 A good alarm

Before delving into what characterizes a good alarm, the start of the interview, asked the candidates to reflect on what an alarm is. The most common definitions included themes like policy violation, signature matching, and notification as a reaction to data monitoring. To create a clearer frame around the term *alarm*, the minimal requirements defined were date and time (timestamp), title, and participant. This work uses the term alarm as a signal that must be dealt with (analyzed). The following paragraphs will discuss different topics the interviewees linked to a good alarm.

General enrichment

Initially, some identified general guidelines will be presented, which are not directly linked to the alarms timestamp, title, or participant attributes.

- **Relevant:** "A good alarm is relevant, meaning that you can act on it.", was one of the opening statements. After some follow-up questions, it seemed that relevance was linked to an operative mechanism, meaning that a trigger should activate a process. The alarm would thereby also need to inform the analyst of why it is relevant.
- **Balance of specific and universal:** An alarm that was in balance with how specific it targeted, and how universal it could trigger was considered favorable. This means that small changes to malware would still trigger an alarm, without triggering similar legitimate software.
- **Interpretable:** The analyst should understand what the alarm is triggering. An example given from one of the interviewees on signature based approaches was "... not a bunch of regexes... split it up, and document each part."
- **Trigger examples:** Each alarm should provide an example of what it tries to trigger, as it is easier for the analyst to compare, rather than making up their attack traffic. A parallel to anomaly could be flow labeled as an attack from the training set.

- **Historical analysis:** Displaying an earlier analysis of the same alarm was argued to help quicken the investigation.
- **Internal trust:** Internal trust in an alarm is improved by seeing its history, and how precise it has been in the past. If the alarm has a history of a high FP rate, the trust factor would lower. From a ML point of view, knowing the model's certainty was deemed useful and important for the candidates. Naturally, a higher certainty percentage would gain more internal trust.
- **External trust:** External trust is not directly connected to the alarm itself, but it can be the author/creator who is known for creating accurate alarms. An example could be that homemade are more trustworthy than those created by a third party.

Alarm title enrichment

The alarm is strict in the sense of its combination of a timestamp, title, and participants. Many of the interviewees reflected on how the title could be split into different categories of interest (COI). To more systematically help the analyst answer two questions that were often presented as fundamental by the candidates, "What is it (the alarm) looking for, and why is that interesting?". One should notice the distinction between understanding the alarm (why it is triggering), and understanding the idea behind why the alarm exists. If an analyst understands why the alarm is triggering, they can act on it regardless of their understanding on its existence. When presented the other way around, it becomes clear that interpreting why an alarm is triggering is the most important of the two.

To support the process, several possible COIs were identified with the goal of providing a clear, well-defined, and easy title.

- **Alarm type:** What kind of system (signature vs anomaly), and what alarm type (e.g. IP, domain) is it generated from?
- **Malware name:** What is the name of the malware the alarm tries to identify.
- **Incident/campaign occurrence:** Perhaps the alarm was a product of an incident within the company, or as part of a larger campaign. An example is the *Log4j* campaign from 2021 [189].
- **CVE:** If the alarm focuses on a Common Vulnerabilities and Exposures (CVE) [13], more information is often available, providing context on what type of systems are vulnerable. An additional link can also be made to the national vulnerability database [43], containing information on severity score an weakness enumeration [191].

[13] The CVE program is a list of publicly disclosed cybersecurity vulnerabilities, where each entry gets a unique ID, often referred to as a CVE number [42].

- **POC:** Closely linked to CVE, is the information of whether or not a Proof of Concept (POC) has been published (Working exploit). Additionally, since the cyber security field is a fast-evolving domain, this paper would recommend a process where the analyst checks if a POC is published when this column is present in the alarm, and a POC is not observed.

- **Exploitation status:** Have the vulnerability been observed actively exploited? If so, then who knows (e.g. company, the cyber security community, everyone)? If it is known to be actively exploited, the same process as recommended in the POC category is applicable here. To check if the status has changed.

- **Tactic and/or techniques:** To quote one of the interviewees, "... if it is C2 we are looking at, then it should be clear in the title.". MITRE ATT&CK® contains a structured list of tactics and techniques used in real events [38].

- **Threat actor:** Which threat actor is the alarm looking for? If that cannot be answered, then perhaps what type (Hobbyist, Hacktivist, Cybercriminal, Advanced Persistent Threat) of the threat actor.

- **Right priority:** Priority is closely associated with the consequence of a vulnerability being exploited. Based on the other categories mentioned, a consequence analysis should be conducted by the analyst creating the signature or anomaly model, such that it is illustrated with the alarm. Examples of priority values could be green (low), yellow (medium), and red (high). An advanced threat actor would contribute to a higher priority.

In total, the categories presented show a clear benefit of the higher steps in the previously presented pyramid of pain (Figure 4).

Alarm participant enrichment

From title enrichment to participant enrichment. Like one interviewee said "the alarm in and of itself is kind of worthless, it is the context (enrichment) that brings value". A claim that could support the adoption of anomaly based methods as long as it is single flow centric (and not evaluating a large batch of flows for each prediction), enrichment is just as easy (or hard) as with the signature based method. As a follow-up question, the candidates mentioning participant enrichment were asked to specify what kind of context would be helpful in a NIDS. The following list emerged:

- **Internal vs external:** The alarm should give a clear view of what is internal traffic, and what is external traffic.

- **Home participant services:** What information is gathered on the company-owned participant, the one the SOC wants to protect. What kind of services are running. If it for example is the company's domain controller, router, or exchange server.

- **Participants records and history:** What kind of external and historical information can be collected to give a better understanding of the alarm? If the participants are represented as IP addresses, DNS, GeoIP [14], information on whether or not the IP has previously been observed, and observed exploiting a CVE, was stated as useful.
- **Home participant sensor placement:** Some of the interviewees expressed the value of knowing where a sensor is placed. In a network setting, the sensor could be placed in front of, on, or behind the firewall, and Network Address Translation (NAT) [15] services. By placing it in front, the SOC may see all connections, even the ones that are not getting into the internal network (e.g. getting blocked by the firewall), however, it might be hard to distinguish devices on the inside. Placing it behind gives transparency to each device, but typical reconnaissance attacks that scan might not be seen. Lastly placing it on the services would bring a *best of both worlds* scenario.
- **Home participant sensor access:** One interviewee mentioned how knowledge of sensor access would be beneficial. An example can be given from a HIDS. Let us say that the system raises an alarm if a specific file does not exist. It would then first of all be important to know if the sensor would have had access to that file in the first place.

Improving the detection and analysis process

As discussed in section 2.2.2, the hypothesis is built up based upon what the analyst sees from the alarm, while goals are formulated from the prediction phase. Two relevant goals the interview focused on was *correlation* and *classification*. The following paragraphs are the candidate's thoughts on it, and how the alarm can support these goals.

Starting with how one alarm can easier be correlated to another alarm, possibly mapping the different kill chain steps to a larger incident, the interviewees made related to the different categories from *title enrichment*. A link to *participant enrichment* was also drawn since by knowing more of the services on the system, the analyst can conclude whether or not two different systems (represented in two alarms) normally influence each other. Alarms on both a web server and a database that commonly shares information would be more relevant to correlate, than a web server and an independent mail server.

The goal of classification brought some unique insights from the interviewees, where one said "... the title in and of itself is not enough for classification. The name, often quickly becoming irrelevant. You (the analyst) need more context information... ". The project sees the statement as a formulation of a possible requirement that the title is easily changed. Other

[14] IP Geolocation tries to geographically map an IP address to a physical location in the real world.

[15] A NAT service maintains a list of different internal IP addresses and ports using an outgoing common external IP address with different ports for each connection. This means that many different devices with different IP addresses can be routed on the internet with the same IP.

candidates mentioned alarm interpretability, severity, and quality in the form of what the alarm wants to detect, and what it detects. The results point to the importance of the previously mentioned enrichment.

5.1.2 Comparing alerts from ML and signatures

The next phase of the interview involved as explained a case, where the candidate studied two alarms on the same technique, one produced by a signature, while the other by an ML based system. The questions resolved around similarities, differences, and how they could be improved.

Signature

As set side by side with ML, the signature felt more detailed and easier to understand. At the same time strict in its capabilities, while giving an isolated view of the data. They are additionally, created and updated in a manual fashion, and one interviewee mentioned how few of the signatures on hosts are both precise and generic.

ML-based possibilities

The interviewees mentioned the following possibilities drawn from their observation of the ML based system section of the case. The model seemed to give a better **overview** of the data flowing to and from the company, creating a stronger connection between data. However one mentioned the need to have a conscious understanding of the **amount of data**, which could result in higher system trustworthiness.

Raw data seems less important than with the signature based system, and the system as a whole was seen as **dynamic and flexible**, slowly changing with the new normal. By looking at added features as compared to signatures, the model was thought to create a more **precise evaluation**. However, with more features also comes a demand for more interpretation from the analyst.

Regarding **severity scoring**, signatures can be graded with a number allowing for a simple prioritization. Similar functionality could easily be given from an anomaly system if the analyst is shown the model's confidence score. Leading to scenarios where the analyst states they "Only have time for everything above 60% today".

Finally, it was mentioned that the ML system "... forces the analyst to better know details in the network.". For example, if a network flow is flagged as *extraction* by the ML system, the analyst would need to check what hides behind both IP addresses. In the process, the analyst discovers that the receiving part of huge chunks of data in fact is a backup service. The analyst now knows a bit more about procedures in the company, and how data flows.

ML-based challenges

Some challenges with the ML system were also expressed. One believed that the system would possibly create **huge variations** in the analysis conducted by different people, possibly demanding more comprehensive processes than with a signature system. Likewise, where should the **boundary** go between malicious and legitimate behavior? A candidate mentioned that "From a *host* perspective ... when the threat actor utilizes legitimate administrator tools instead of malware, where should the line go between lawful and unlawful use.". This kind of process would have to be created in cooperation with both the analyst and the data engineer.

Another challenge, due to the fact that the data flowing in each sensor is unique, a model might need to be **trained** for each sensor. This adds a large layer of complexity for the analysts, if not properly managed. The model were also perceived as giving **little control** to the decisions being made, reflecting to the concept of a black-box as recognized by an analyst. Not knowing how to influence and change its predictions. Such that a form of **trust** would have to be established, as mentioned earlier. This trust might only build over time as the analyst develops an understanding of the score for each model.

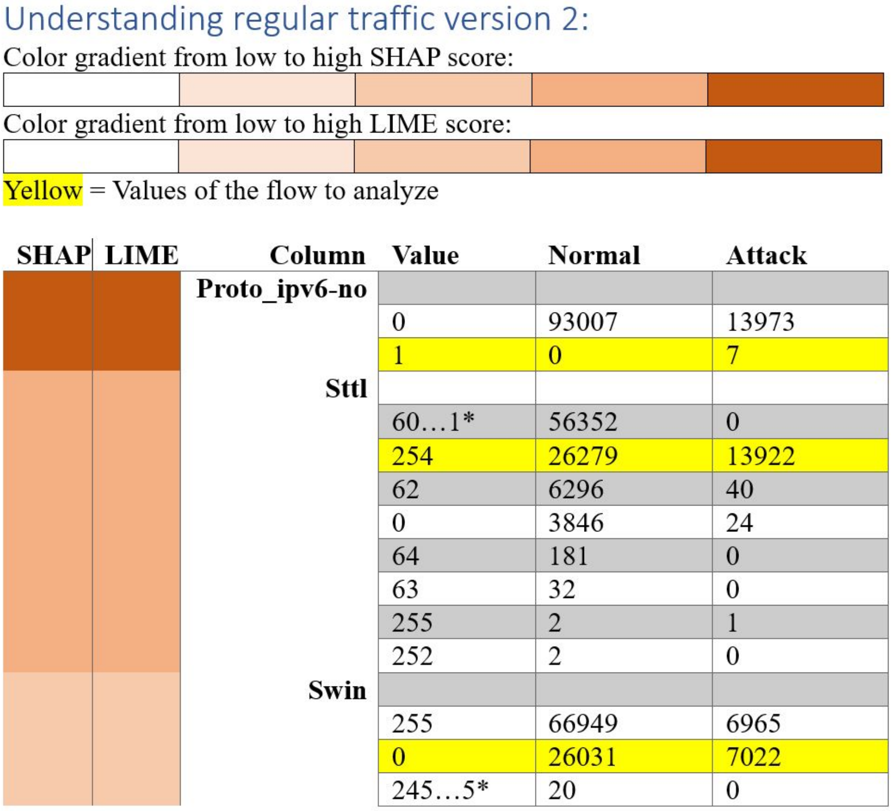
The use of XAI

The following paragraphs focus on how the interviewees experienced the usage of XAI in the ML system. First, some general impressions are presented, before comments on features, and the XAI methods scores are examined. Nonetheless, a great majority of the candidates found the methods useful, and was even seemed by some as a necessity if the system is based on a black-box model.

Since the model in the example uses more features than what might be normal for a signature, the candidates classify the analysis process as more advanced, demanding a higher technical understanding of most of the features present. They also found it practical that the methods identified important features. To quote one candidate "They (SHAP and LIME) show values (features) I never would have thought of checking ... speeding up my work, since I can start with the most important ones". In that way, it also helps to remove noise. Additionally, the interview sheet only showed the top three features generated from SHAP and LIME. A small number of the interviewees expressed a wish to see all the values, to better understand their meaning.

Seeing all the values is not the only suggestion from the candidates. Some wanted to replace the numbers with graphs, colors, or normalized numbers. Since "... an analyst would have little insight into whether or not 0,18 (score given by both SHAP and LIME to the column *proto ipv6-no*) is a high or a low number". An interesting proposal was to remove the XAI methods tables completely, and instead show shades of a color in the *understanding regular traffic* (Appendix B.5) table was discussed. The values of which the flow in question contained could also be highlighted.

The feedback was used to create a new table, seen in figure 10.



Understanding regular traffic version 2:

Color gradient from low to high SHAP score:

Color gradient from low to high LIME score:

Yellow = Values of the flow to analyze

SHAP	LIME	Column	Value	Normal	Attack
		Proto_ipv6-no			
			0	93007	13973
			1	0	7
		Sttl			
			60…1*	56352	0
			254	26279	13922
			62	6296	40
			0	3846	24
			64	181	0
			63	32	0
			255	2	1
			252	2	0
		Swin			
			255	66949	6965
			0	26031	7022
			245…5*	20	0

Figure 10: Proof of concept, of how the *Understanding regular traffic* table could look, based on feedback from the interview. The score from SHAP and LIME is shown as a color gradient, the values of the flow an alarm have triggered on is highlighted with yellow, and the table is sorted on the column *normal*, instead of *value*.

Global interpretability

One participant mentioned that "I want to see if the same values (features) is present in the global assessment (global interpretability)". Traditionally global interpretability was only deemed relevant for data scientists that wanted to debug and improve their model. This surprising statement shows that value might be drawn from an additional XAI field.

Understanding regular traffic

Maintaining focus on the last table in the case *understanding regular traffic* (Appendix B.5), the candidates felt that it gave a clearer view of how normal traffic looked, "... something I (the analyst) can use a lot of time on with data analytic tools". The table seemed to make the analytical process easier. A request to sort the table on numbers in the *normal* column, instead of *values* was mentioned, as that might make it simpler to see if something is normal. This feedback was also included in the new proof of concept figure 10.

Textual description

The ML model was also presented with a textual description as discussed in section 3.2.4. Every candidate had a positive encounter with the description, especially since it shows if something has never been seen before, as in the use case presented where 0% of all legitimate have never used *ipv6-no*. Some interviewees meant that the description could make it easier for inexperienced analysts and that it gave a "... nice quick overview of the situation.", leading to the creation of a hypothesis.

Some also suggested changes to the description. One change was to also have the score of each feature given from the XAI methods as well. However as previously discussed, not necessarily just the score, but simplified, perhaps in the form of a graph. Additionally, percents and features could be made **bold**.

Machine learning supporting signature

The candidates reflected on how the two systems (ML and signature) could co-exist and assist each other. Flow from triggered alarms based on signatures could be used as labeled data for a ML model. The other way around, anomalies could detect both new previously unseen attacks (zero days) with the potential of creating new signatures, as well as find untraditional feature candidates to make a signature more precise. Thus resulting in a possibly better alarm.

It was also mentioned that by using both systems if an alarm is generated by a signature, the flow could potentially be fed into the ML model, thereby getting a prediction score, and with XAI insight into important features for that prediction. Another scenario is to increase priority if both systems classify an alarm as malicious, thereby supporting the analyst in prioritization.

Finally, by looking at alarms over time. If an IP that shows up in the signature system has previously triggered in the ML model, and at that time, it was not further investigated since the event was not considered malicious, then that would now help the analyst build a better case to further the investigation.

Method deficiency

The candidates found potential for improvement. As mentioned earlier, getting more context data, seeing earlier analysis reports on the same alarm, getting the models prediction score, making the XAI score more visual, and seeing example traffic of similar art of which the signature or ML model is trying to detect, was uttered.

The interviewees wanted a function giving them a better view of the amount of data being transmitted over time. This could be a graph showing the amount of data or number of packets each hour in a day, and/or for every day in the week. Additionally, a function that automatically assesses similarity between example (historic) flow, and the flow triggered, would give an indication of whether or not the supposed malicious traffic is *common* or not previously seen.

5.1.3 Anomalous vs Malicious

An anomaly system will have a fundamental difference as compared to signatures, because the alarm is derived from the abnormal activity, while signatures base their origin on a known malicious behavior. When asked to reflect on the matter, the candidates did not see it as a big challenge for the analyst. A short brief on the matter of anomaly, along with clear labeling on the alarms would go a long way. To quote one on the matter "Even if an anomaly is not malicious, it could show a weakness in the system that should be handled.".

5.1.4 Alarm per IP functionality

Since one could argue that anomaly system, by analyzing and using data that can span over many months, have a wider and, in the case of deep neural networks, a more complex relationship to time than signatures, extra functionality that focuses on tracking aspects over time might be more useful with an anomaly system. The interviewees had some thoughts regarding a function that tracks the number of alarms generated per *marked* IP. Think of a situation where an alarm is generated, but the information is too vague to escalate, so the supposed attacking IP is added to a *watchlist*. When enough alarms have been triggered on a watchlist IP, the alarm is escalated.

First of all, it seemed useful as a way to identify malicious behavior before an incident occurs, making it preventive. IPs in the watchlist could also automatically add a higher priority to alarms of which they are present. A part of the process is when an alarm with a watchlist IP is activated, the analyst would make a new assessment if any new information on the IP has surfaced. Again comes the suggestion of a graph instead of just numbers, that can show the number of alarms the IP was present over time.

5.1.5 Statement rating

As part of the ending in section 4.4, the candidates were asked to grade three questions low, medium, and high. (1) Firstly is how they felt that each system contributed to alarm interpretability. As can be seen from the results shown in Figure 11, the two systems are quite similar. One interviewee also commented that the *textual description* was important in their rating.

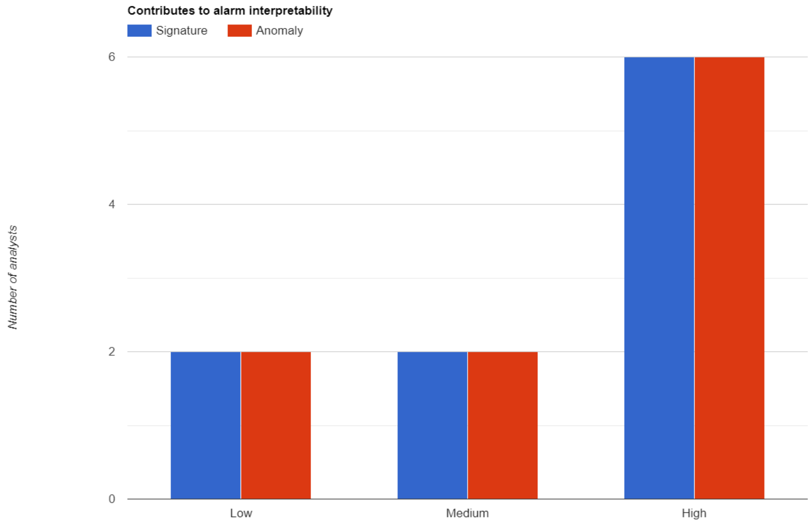


Figure 11: Graph showing the number of candidates (10 total) meaning how each system (signature and ML) contributed to alarm interpretability

The second question revolved around (2) how they felt that each system contributed to shortening the analysis time. This research would argue that the graph shown in Figure 12 is depicting the systems as more alike than different. This is interesting since the ML system did provide some more information that the analyst would have to consider. However, one comment on the matter might give some insight to why: “The model (ML) shows more information, thereby demanding more time, however, the analysis would be faster.”.

The third question dives into what is needed from the analyst, asking (3) how much they felt that each system demanded in the form of knowledge and intelligence from the analyst. The results, as shown in Figure 13, point out that the ML system used in this thesis does require a little more from the analyst, compared to a more traditional signature. A comment on the matter mentioned the high number of variations (features) in the ML data as a reason to give the system a higher rating.

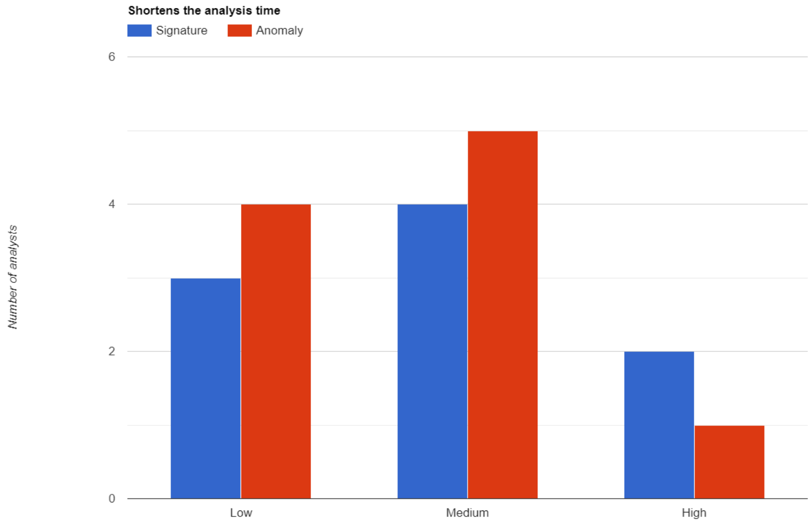


Figure 12: Graph showing the number of candidates (10 total) meaning how each system (signature and ML) contributed to shortening the time spent analyzing each alarm

5.1.6 Advice to an anomaly system developer

Concluding the interview, the candidates were asked to give some advice to a developer creating an anomaly based system for a SOC. Many tips were already mentioned earlier in this result section, however, the following list contains additional suggestions:

- **Customisability:** The system should support customizability from the SOC, such that extra functions and how each function is shown can be altered.
- **No abbreviations:** The system should not use unnormal abbreviations like in the interview's case where *swin* is an abbreviation of *source windows size value*.
- **Neutral data assessment:** The developer should be neutral in its data assessment since the correlation might not be obvious.
- **Use colors and values:** Use colors smart, however, do not underestimate the analyst, try to show as much detail as possible, a simple pop-up for further information would help.

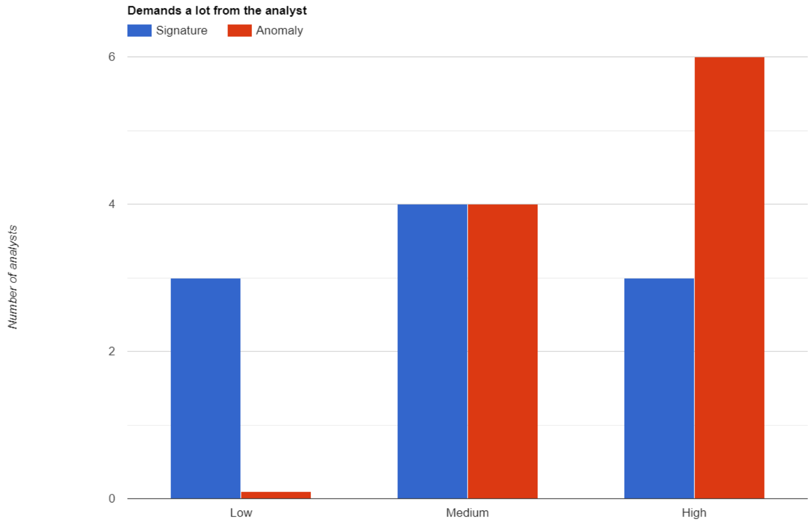


Figure 13: Graph showing the number of candidates (10 total) meaning how much each system (signature and ML) demands from an analyst before beein properly used

- **System co-existence:** Both the anomaly and signature should be handled in the same tool, and ideally handled by the same analyst.
- **Modular platform:** Make the base platform modular, and do not try to create every functionality in one go. Instead iteratively publish one and one, gathering feedback from analysts before moving to the next. "Many projects try to start from nothing, and go to bleeding edge".

5.2 Discussion

The results show a possibility to better educate the analysts on the link between features and the attacking technique, with the use of XAI. The same has been claimed in similar research using SHAP [196].

Another point is what was earlier mentioned in section 5.2.1, concerning the correlation between education or experience and the use of ML tools discussed in [137]. Even if the interviewees were not observed using a tool, a comparison can be drawn as to who deemed the solution demanding and not. A brief analysis of the matter can not find a correlation between the two data points. Given the small amount of 10 candidates, one could say that the results only weakly indicate in favor of earlier research.

The list of advice to a system developer, and as mentioned with the difficulty of understanding the XAI score, can also be seen in other research. In [137], one of their recommendations was to provide the analyst with thorough knowledge on how to understand (in their case) the ML output, similarly presenting pop-ups as a solution. Other research also recommends including the end-user during development, such that interpretability is guaranteed [8, 30].

The results also show that the candidates want a more conscious understanding of the ML models prediction score. The same can be seen from other research [137]. However when providing the score, one of their analysts gained a mistrust when questioning the model's predictions, asking "Why trust this score?". The project and the cited research indicate that more information should be provided as to how a prediction is reached, and perhaps XAI is part of that solution. Leading to efficient incident response.

Since this research discovered that earlier research identified a difficulty for analysts to understand the difference between what is a malicious alarm, and what can only be considered anomalous [137] (which is the case for anomaly based systems). It was deemed natural to include it as a question to the candidates with the goal of possibly finding extra characteristics one could add to improve alarms. However as can be seen in section 5.1.3, this was not seen as a problem. Since this project only conducted interviews, while earlier research also saw their analysts use a specific tool, it might be the case that what seems easy in theory is in fact harder to analyze in practice.

Section 2.6.1 presented similar research focusing on the technical utilization of XAI methods in anomaly detection. In the case of Karn et al. [95], it would be interesting if they elaborated on how the administrator used the explainable output from the XAI methods (LIME, SHAP, and an autoencoding technique for RNN) to decide on whether or not to take action on a possibly cryptominer-infected host.

Furthermore, Mane and Rao's [115] paper uses both a global post-hoc XAI method and a local post-hoc method on a ANN intrusion detection system. The global for a data scientist to debug and improve the model, and the local for the end-user to understand feature contributions. The system also shows representative examples from the training data. In relation to this research, it would be relevant to merge the security analyst and end-user roles, combining a data example with an explanation. The same also applies here, that more details on how the information was made available would be highly relevant.

5.2.1 Explainabilities impact in a SOC

The following section will highlight the effect XAI methods could have in a SOC, by using a typical framework from research describing the different operations and functions in a SOC environment. These are PPT [140, 114, 182]. Here, *people* can be the ones handling the alarms as described in a *process* (step-by-step guide), with the provided *technology* (tools). Figure 14 shows how the areas influence each other.

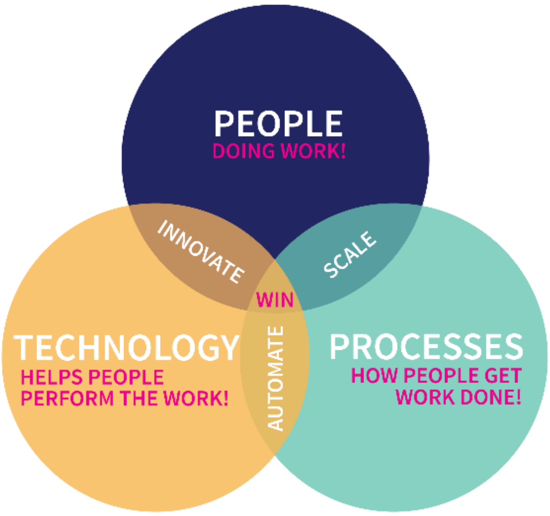


Figure 14: Model depicting how PPT coexist and affect each other [1].

People

Starting with the most valuable asset, Oesch et al. [137] identified some noteworthy observations during their research on SOC analysts' usage of two ML tools. The study "... found no correlation between analysts' level of education or years of experience and their performance with either tool ...". In that sense, they suggest that other components might be influential, like prior background knowledge and personality. The results indicate a need to educate users on the specific tool used when it comes to ML. This is further supported as the same paper found a lack of understanding regarding how the tools generated scores, adding to the performance issues of tool misuse and distrust.

Another question, which is not directly linked to explainability, but could be affected by it, is whether or not the use of anomaly detection systems, and DNN is a natural candidate to base such a tool on. Goodall et al. [66] found in their study on anomaly detection and information visualization that the biggest challenge security analysts faced when using their system (named "Situ"), was a change of habit in the analyst's mindset. This was especially important when handling an anomaly alarm as just an anomaly, and not malicious (since an anomaly is not necessarily malicious). These analysts was used to an IDS based on signatures, where every signature originates from a considered malicious behavior. This research indicates that more transparency as a result of XAI methods would decrease the barrier to adopt anomaly detection systems, if the methods used eases the analysis work by showing important features.

Process

Moving to how processes might be impacted, XAI could be used to direct more focus to anomalies, increasing knowledge of the observed system. Creating new procedures as greater insight, which earlier demanded much more work is achieved. A short example might be that in pre-explainability, the analyst would have to use more time analyzing an anomaly generated as the result of uncommon header combinations in a transport layer protocol, simply because the defined process of such work would start at the lower network layers, working upwards. However, that might shift as post-explainability could point the analyst directly to the transport layer header. Thereby increasing efficiency.

Secondly, when discussing processes, it would be relevant to consider how explainability impacts hypothesis creation (as presented in section 2.2.2) when comparing signatures and anomalies. A non-complex signature could be enough for the analyst to build a sufficient hypothesis. For example, picture a signature specifying that if 100 connections are made from the same source to a login page in 1 second, raise an alarm. The analyst may interpret it as brute force, making predictions on the next steps in the investigation, like checking if any of the connections bypassed the login. However anomaly-based alarms from a DNN lack the trait of possibly providing inherent explainability. This highlights an important difference between the two groups, where one *in nature* (without any supporting tools like XAI) has the possibility of contributing to the hypothesis, while the other does not.

The results in this thesis show that LIME and SHAP, by presenting influential features does contribute to an analysts hypothesis. Firstly in providing interpretability (Figure 11), but also in giving insight to aspects like system weaknesses (see section 5.1.3), making it possible to create different hypothesis's than a signature-based method.

Technology

From a technological aspect, XAI could influence the trust an analyst would have in the tool, by giving insight into why the model predicts the way it does, thus responding to one of the identified challenges by summer & paxons [178] regarding lack of model explainability. However, the optimal visual solution to, for example, portray the score each feature holds, seems to still be an open issue. Possible solutions could be as a graph, or color-coded.

The technology's possible effect on data is also interesting. When dealing with an alarm, the analyst might only look at header values (features) that are familiar and logically important regarding a given tactic. On the other hand, XAI could point out other relevant features in the data the ML model has identified, thanks to its big data processing power. The use of new knowledge that identifies tactics is not isolated to new data. Historical data should also be accounted for as a subject for investigation. The technology can by that be considered a knowledge enhancer.

5.2.2 LIME and SHAP impacting the result

Section 2.5.5 presented some advantages and disadvantages of LIME and SHAP. These are presented and impact discussed:

- **LIME - Instability:** LIME's instability related to explanation of similar/close data points is mitigated by comparing the results of LIME, to SHAP.
- **SHAP - Need of data and feature independence:** The need SHAP have for data access in order to make a prediction will not influence this work, since a dataset is provided and even openly available. This could on the other hand have an effect on usage within often restricted and closed SOC environments and should be considered before operational use.
- **SHAP and LIME - Picking non-representable data:** Both SHAP and LIME can create non-representable data (also known as *false samples*) for their analysis. The project does not have a mitigation measure towards the topic.
- **SHAP and LIME - hiding bias:** Bias might be hidden from LIME and SHAP, thereby creating misleading interpretations. Since one of the preliminaries for hiding bias resolves around "an adversary with an incentive to deploy a biased classifier", and this project focuses on the possible usage of the methods, thereby not gaining anything on hiding biases in the data, the disadvantage is considered to be of little importance. However, the end-user would need to trust the data scientist. The analyst, as the receiving part of a LIME and SHAP explanation, could therefore question the output's truthfulness. The field of resolving around creating adversarial robust explanations is still a work in progress.

5.2.3 Research bias

Regarding how the researcher's bias might have affected the data, it is possible that some of the candidate's words were unconsciously disregarded because it was understood as unclear speech, when in fact it was not given enough consideration because it did not fit with the preconception. However since the interview was recorded, it was possible to analyze the data multiple times, while being consciously aware of the possibility of preconceptions, this work is therefore considered to have handled the risk of bias well enough.

This section have described how the results compare to other similar research in the field. The research found both support and contradiction. A greater discussion on XAI's impact for a SOC in a PPT framework was also given. Finally, the challenges of LIME and SHAP's impact on the result, and a comment on researchers bias was examined.

6 Conclusion

The objective of this thesis was to find characteristics of a good alarm from a SOC's point of view while identifying requirements for a ML-based alarm system. Additionally, exploring how XAI can achieve that and be interacted with. That was done, first by conducting an experiment to test out XAI interaction and functionality, before carrying out an interview with SOC analysts. The interview used the outcome from the experiment to gather data on the use of XAI.

Starting with the first research question **Q.1**, the results documented in chapter 5 first of all show a general belief that alarms can be characterized into good and bad. Where good alarms, make it clear to the analyst what they are trying to detect (with examples), and why that is important while being universal, precise, and trustworthy. To answer those tasks, the project identified and organized different attributes of enrichment into three categories: General, title, and participant. Showing both detailed values and functionality a SOC could use to improve the detection and analysis process.

Secondly regarding research question **Q.2**, many of the same enrichments guidelines for a good alarm can be utilized by ML. Other requirements was to give a conscious understanding of the amount of data, a severity scoring based on the models prediction confidence, and a clear process description on how to handle the alarms to prevent huge analysis variations. Some identified challenges that need solving is a way to manage many models trained on different sensors, and establish model trust.

Another topic explored was how the ML model and signature based system complemented each other via generating labeled data and (with XAI) identifying important features, while at the same time providing unique insight and support to an analyst through shorter analysis time, clearer normal traffic view, and better prioritization.

Finally for research question **Q.3**, the way XAI was used and presented based on the experiment was also well received. The rating in section 5.1.5 indicates that LIME and SHAP along with the *regular traffic table*, and *the textual description* added value in the form of interpretability and a shorter analysis time, supporting characteristics of a good alarm. The suggested changes to both the functions and platform are very manageable (see figure 10), and will hopefully lower the threshold of usage. Additional functionality like *alarms per IP*, was also deemed useful with small adjustments.

6.1 Future work

The work in this thesis have made the first steps towards connecting the XAI field to both cyber security and human interaction. The project have shown the importance of working closely with a security analyst when advancing data driven intrusion detection systems. Future work on the matter is recommended to derive from an open-source collaborative project, ideally involving academia and security actors in both the public and private sectors. Where the base is a modular platform that easily enables alarm enrichment. Different anomaly and signature based systems can thereby be integrated and experimented with. It would be natural to start with a survey on existing projects that match the criteria found in this thesis.

Some of the identified functions in this work that could provide a positive effect on such a platform are mentioned in section 5.1.2. One is **example traffic**. A project named *ProtoDash* seem to be highly relevant in this scenario [73], given that it is a fast prototype (data point) selection method. By finding datapoints that best represent the one in the alarm, would as seen from the results ease the analysts work.

Another interesting function would be to automatically generate proposed signatures based on an anomaly output. The functions pipeline would also check historically how well the new signature could behave while letting the analyst specify how generic (perhaps in the form of fewer specified features) the signature should be.

Additionally, in section 3.2.3 in relation to the ML model, the demand for a continuous data stream is mentioned, alongside inspection of historic data. Further work could alongside these add the possibility of continuous model improvement/training based on both analyst response and shifts in the user's normal traffic habits.

Future research may change individual variables to identify new characteristics for a good alarm, thereby steering XAI requirements. Some of the factors are:

- **XAI methods:** As described in section 2.5 on XAI, the field is continuously evolving. It would be interesting to see how different methods may add new insights while mitigating the discussed risk of *false samples* 2.5.5. regarding model-specific methods for ANN [188], the project could not find an easy to implement method that could be considered relevant to an analyst with a focus on tabular data, however it would be interesting to see the possibilities they can create for an analyst. Other methods are counterfactual explanations [128], intrinsic methods like decision trees, knowledge graphs [104, 103], as well as a look into language models like the InstructGPT [141], could prove useful. Additionally, the results did show a desire for global interpretability, so methods specified in that field, along with how they are visualized to the analyst could be significant.
- **Techniques:** This research focused on the accessible concept of reconnaissance. Other techniques might have different enrichment needs which were not identified. Different and bigger (in the form of various aspects of the same attack) techniques should be tested.

- **End user:** Different end users might have other concerns than a SOC analyst (This also applies to different needs between a level 1 and level 2 analyst). Since this thesis is in cooperation with FFI, a natural example of another end user would be military personnel. Future research might find a larger need for interpretability if the end-user does not have a significant portion of technical and incident response experience.
- **SOC environment:** By investigating different SOC environments, generalization could be easier as a larger amount of PPT combinations are included. The joined insight would in greater lengths strengthen research in the field.
- **Domain:** Even though individuals with a heavily host-based profession were represented in the interview, a clear majority had the most experience with networks. Future work should take the different domains into account.
- **ML model architecture:** Machine learning is a vast field, and a perfect configuration was not the scope in this thesis, however parts of the result in this research is depend on a operationalized model. It is therefore relevant for future research to experiment with different architectures, at least changing both the depth (number of layers) and width (number of nodes).

This project would also urge future research to not only experiment with different XAI methods, but also how the methods can be made more explainable. An example from the interview case in this thesis shows that the feature *proto_ipv6-no* was important (Appendix B.4). By asking *why* that feature might be important, and how that can be displayed to an analyst, this project created the *regular traffic* table in appendix B.5, as well as a textual description (Appendix B.4). The researcher concludes that these types of functionalities will be central in an operationalized setting, and thereby important in future work.

Regarding research design, future work could utilize the case studies observational research method [127] on the XAI featured platform, along with an interview afterward to clarify and investigate interesting findings. Hopefully reaching a state where the platform is completely operationalized, and where innovation is backed by research. Other common methods future research could consider [146, 145]:

- **Surveys:** Questionnaires that can be given to security analysts, containing open-ended questions.
- **Observations:** Observing and recording how the security analyst might use an anomaly based tool with XAI capabilities.
- **Secondary data:** Gather documents/playbooks that specifies analysts' processes, or images and videos that could show the analysis process.

- **Focus groups:** Asking questions to a group of security analysts, facilitating discussion.
- **Interview:** Asking questions in a one-to-one conversation with the analyst.
- **Experiment:** Conducting tests on how different XAI methods could be combined with an alarm generating black-box ML model to showcase the output to an analyst.

Future work should also to a greater extent investigate different topics in Cyber Situation Awareness (CSA) like visualization and user-centered design concepts, human-computer interaction, as well as topics for Human-Automation Interaction (HAI) within cyber security, like Human-automation "Teaming". A more general approach as done in the field of Human Computer Interaction (HCI) [136], might also be relevant to create an efficient and seamless integration of tools, processes, and the analyst.

References

[1] Melih Abdullah. *Deming cycle - Structure of continuous improvement*. en-US. Nov. 2020. URL: https://melih.com/continually-improve-structured-approach-deming-cycle/ (visited on 01/13/2022).

[2] Oludare Isaac Abiodun et al. "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4.11 (Nov. 2018), e00938. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2018.e00938. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6260436/ (visited on 04/20/2022).

[3] S Agatonovic-Kustrin and R Beresford. "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research". en. In: *Journal of Pharmaceutical and Biomedical Analysis* 22.5 (June 2000), pp. 717–727. ISSN: 0731-7085. DOI: 10.1016/S0731-7085(99)00272-1. URL: https://www.sciencedirect.com/science/article/pii/S0731708599002721 (visited on 03/03/2022).

[4] Afaq Ahmad, Shashwat Ganguly, and Fan Wang. "Optimised building energy and indoor microclimatic predictions using knowledge-based system identification in a historical art gallery". In: *Neural Computing and Applications* 32 (Apr. 2020). DOI: 10.1007/s00521-019-04224-7.

[5] Ayoub Si-Ahmed, Mohammed Ali Al-Garadi, and Narhimene Boustia. *Survey of Machine Learning Based Intrusion Detection Methods for Internet of Medical Things*. Tech. rep. arXiv:2202.09657. arXiv:2202.09657 [cs] type: article. arXiv, Feb. 2022. DOI: 10.48550/arXiv.2202.09657. URL: http://arxiv.org/abs/2202.09657 (visited on 05/22/2022).

[6] Tarem Ahmed, Mark Coates, and Anukool Lakhina. "Multivariate Online Anomaly Detection Using Kernel Recursive Least Squares". In: *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*. ISSN: 0743-166X. May 2007, pp. 625–633. DOI: 10.1109/INFCOM.2007.79.

[7] Tariq Ahmed et al. "Reduction of Alert Fatigue using Extended Isolation Forest". In: *2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*. Vol. 1. Dec. 2021, pp. 1–5. DOI: 10.1109/FABS52071.2021.9702617.

[8] Olusola Akinrolabu, Ioannis Agrafiotis, and Arnau Erola. "The challenge of detecting sophisticated attacks: Insights from SOC Analysts". In: *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ARES 2018. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 1–9. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233280. URL: https://doi.org/10.1145/3230833.3233280 (visited on 05/14/2022).

[9] Bio Akram and Dion Ogi. “The Making of Indicator of Compromise using Malware Reverse Engineering Techniques”. In: *2020 International Conference on ICT for Smart Society (ICISS)*. Vol. CFP2013V-ART. 2020, pp. 1–6. DOI: 10.1109/ICISS50791.2020.9307581.

[10] Mohammed Alasli and Taher Ghaleb. “Review of Signature-based Techniques in Antivirus Products”. In: Apr. 2019, pp. 1–6. DOI: 10.1109/ICCISci.2019.8716381.

[11] Alex Hinchliffe. *DNS Tunneling: how DNS can be (ab)used by malicious actors*. en-US. Mar. 2019. URL: https://unit42.paloaltonetworks.com/dns-tunneling-how-dns-can-be-abused-by-malicious-actors/ (visited on 05/20/2022).

[12] Jan Harald Alnes. *falsifikasjon – vitenskapsteori*. nb. Dec. 2021. URL: http://snl.no/falsifikasjon_-_vitenskapsteori (visited on 02/16/2022).

[13] Riyad Alshammari and A. Nur Zincir-Heywood. “Machine learning based encrypted traffic classification: Identifying SSH and Skype”. In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. ISSN: 2329-6275. July 2009, pp. 1–8. DOI: 10.1109/CISDA.2009.5356534.

[14] Aakash Alurkar et al. “A Comparative Analysis and Discussion of Email Spam Classification Methods Using Machine Learning Techniques”. In: May 2019, pp. 185–206. ISBN: 978-0-429-44095-3. DOI: 10.1201/9780429440953-10.

[15] David Alvarez-Melis and Tommi S. Jaakkola. “On the Robustness of Interpretability Methods”. In: *arXiv:1806.08049 [cs, stat]* (June 2018). arXiv: 1806.08049. URL: http://arxiv.org/abs/1806.08049 (visited on 04/13/2022).

[16] Thiago Alves, Rishabh Das, and Thomas Morris. “Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers”. In: *IEEE Embedded Systems Letters* 10.3 (Sept. 2018). Conference Name: IEEE Embedded Systems Letters, pp. 99–102. ISSN: 1943-0671. DOI: 10.1109/LES.2018.2823906.

[17] Aditya Amberkar et al. “Speech recognition using recurrent neural networks”. In: *2018 international conference on current trends towards converging technologies (ICCTCT)*. IEEE. 2018, pp. 1–4.

[18] Marco Ancona et al. “Gradient-Based Attribution Methods”. en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 169–191. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_9. URL: https://doi.org/10.1007/978-3-030-28954-6_9 (visited on 04/02/2022).

[19] Ann-Kristin Elstad. “Critical Success Factors When Implementing an Enterprise System - An Employee Perspective”. en. PhD thesis. Norges Handelshøyskole, 2014. (Visited on 05/04/2022).

[20] DARPA BAA. *Broad Agency Announcement Explainable Artificial Intelligence (XAI)*. en-US. Aug. 2016. URL: `https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf` (visited on 01/22/2022).

[21] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". en. In: *Information Fusion* 58 (June 2020), pp. 82–115. ISSN: 1566-2535. DOI: `10.1016/j.inffus.2019.12.012`. URL: `https://www.sciencedirect.com/science/article/pii/S1566253519308103` (visited on 01/19/2022).

[22] David Bau et al. "Network Dissection: Quantifying Interpretability of Deep Visual Representations". In: *Computer Vision and Pattern Recognition*. 2017.

[23] David Bau et al. "Understanding the role of individual units in a deep neural network". In: *Proceedings of the National Academy of Sciences* (2020). Publisher: National Academy of Sciences. ISSN: 0027-8424. DOI: `10.1073/pnas.1907375117`. URL: `https://www.pnas.org/content/early/2020/08/31/1907375117`.

[24] Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. "Network Anomaly Detection: Methods, Systems and Tools". In: *IEEE Communications Surveys Tutorials* 16.1 (2014). Conference Name: IEEE Communications Surveys Tutorials, pp. 303–336. ISSN: 1553-877X. DOI: `10.1109/SURV.2013.052213.00046`.

[25] Leyla Bilge and Tudor Dumitraş. "Before we knew it: an empirical study of zero-day attacks in the real world". In: *Proceedings of the 2012 ACM conference on Computer and communications security*. CCS '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 833–844. ISBN: 978-1-4503-1651-4. DOI: `10.1145/2382196.2382284`. URL: `https://doi.org/10.1145/2382196.2382284` (visited on 02/03/2022).

[26] Jason Brownlee. *How to Avoid Overfitting in Deep Learning Neural Networks*. en-US. Dec. 2018. URL: `https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/` (visited on 05/24/2022).

[27] Jason Brownlee. *Loss and Loss Functions for Training Deep Learning Neural Networks*. en-US. Jan. 2019. URL: `https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/` (visited on 05/24/2022).

[28] Bushra A. Alahmadi and Louise Axon. "99% False Positives: A Qualitative Study of {SOC} Analysts' Perspectives on Security Alarms". en. In: 2022. URL: `https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi` (visited on 01/22/2022).

[29] Alfredo Carrillo, Luis Fernando Cantú, and Alejandro Noriega. "Individual Explanations in Machine Learning Models: A Survey for Practitioners". In: *CoRR* abs/2104.04144 (2021). arXiv: 2104.04144. URL: `https://arxiv.org/abs/2104.04144`.

[30] Dylan Cashman et al. “A User-based Visual Analytics Workflow for Exploratory Model Analysis”. In: *Computer Graphics Forum* 38.3 (June 2019). arXiv:1809.10782 [cs], pp. 185–199. ISSN: 0167-7055, 1467-8659. DOI: 10.1111/cgf.13681. URL: http://arxiv.org/abs/1809.10782 (visited on 05/14/2022).

[31] Davide Castelvecchi. “Can we open the black box of AI?” en. In: *Nature News* 538.7623 (Oct. 2016). Cg_type: Nature News Section: News Feature, p. 20. DOI: 10.1038/538020a. URL: http://www.nature.com/news/can-we-open-the-black-box-of-ai-1.20731 (visited on 01/19/2022).

[32] Yu-Liang Chou et al. “Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications”. en. In: *Information Fusion* 81 (2022), pp. 59–83. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2021.11.003. URL: https://www.sciencedirect.com/science/article/pii/S1566253521002281 (visited on 01/19/2022).

[33] Paul Cichonski et al. *Computer Security Incident Handling Guide*. en. Aug. 2012. DOI: https://doi.org/10.6028/NIST.SP.800-61r2.

[34] CISA. *Traffic Light Protocol (TLP) Definitions and Usage | CISA*. URL: https://www.cisa.gov/tlp (visited on 05/04/2022).

[35] Cisco. *Snort - Network Intrusion Detection & Prevention System*. URL: https://www.snort.org/ (visited on 03/08/2022).

[36] Michael D. Coovert, Rachel Dreibelbis, and Randy Borum. “Factors Influencing the Human–Technology Interface for Effective Cyber Security Performance”. In: *Psychosocial Dynamics of Cyber Security*. Num Pages: 24. Routledge, 2016. ISBN: 978-1-315-79635-2.

[37] B.J. Copeland. *artificial intelligence | Definition, Examples, Types, Applications, Companies, & Facts | Britannica*. en. Dec. 2021. URL: https://www.britannica.com/technology/artificial-intelligence (visited on 01/13/2022).

[38] The MITRE Corporation. *MITRE ATT&CK®*. 2022. URL: https://attack.mitre.org/ (visited on 05/10/2022).

[39] Kelton A. P. da Costa et al. “Internet of Things: A survey on machine learning-based intrusion detection approaches”. en. In: *Computer Networks* 151 (Mar. 2019), pp. 147–157. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2019.01.023. URL: https://www.sciencedirect.com/science/article/pii/S1389128618308739 (visited on 05/22/2022).

[40] Critical Start. *In Cybersecurity Every Alert Matters - an IDC White Paper*. en-US. URL: https://www.criticalstart.com/resources/in-cybersecurity-every-alert-matters/ (visited on 05/02/2022).

[41] Christopher Crowley and John Pescatore. *A SANS 2021 Survey: Security Operations Center (SOC) | SANS Institute*. Oct. 2021. URL: https://www.sans.org/white-papers/sans-2021-survey-security-operations-center-soc/ (visited on 04/24/2022).

[42] CVE. *Overview | CVE*. URL: https://www.cve.org/About/Overview (visited on 05/09/2022).

[43] CISA Cybersecurity & Infrastructure Security Agency. *NVD - Home*. URL: https://nvd.nist.gov/ (visited on 05/25/2022).

[44] Emmanuel Gbenga Dada et al. “Machine learning for email spam filtering: review, approaches and open research problems”. en. In: *Heliyon* 5.6 (June 2019), e01802. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2019.e01802. URL: https://www.sciencedirect.com/science/article/pii/S2405844018353404 (visited on 05/22/2022).

[45] David J Bianco. *The Pyramid of Pain*. en. URL: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html (visited on 02/02/2022).

[46] Shuchisnigdha Deb and David Claudio. “Alarm fatigue and its influence on staff performance”. In: *IIE Transactions on Healthcare Systems Engineering* 5.3 (July 2015). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/19488300.2015.1062065, pp. 183–196. ISSN: 1948-8300. DOI: 10.1080/19488300.2015.1062065. URL: https://doi.org/10.1080/19488300.2015.1062065 (visited on 05/24/2022).

[47] Alfred DeMaris. “A Tutorial in Logistic Regression”. In: *Journal of Marriage and Family* 57.4 (1995), pp. 956–968. ISSN: 00222445, 17413737. URL: http://www.jstor.org/stable/353415.

[48] Karthika Renuka Dhanaraj et al. “Spam Classification Based on Supervised Learning Using Machine Learning Techniques”. In: *ICTACT Journal on Communication Technology* 2 (July 2011). DOI: 10.1109/PACC.2011.5979035.

[49] Jake Drew, Michael Hahsler, and Tyler Moore. “Polymorphic malware detection using sequence classification methods and ensembles”. In: *EURASIP Journal on Information Security* 2017.1 (Jan. 2017), p. 2. ISSN: 1687-417X. DOI: 10.1186/s13635-017-0055-6. URL: https://doi.org/10.1186/s13635-017-0055-6.

[50] CSRC Content Editor. *availability - Glossary | CSRC*. EN-US. URL: https://csrc.nist.gov/glossary/term/availability (visited on 05/20/2022).

[51] CSRC Content Editor. *confidentiality - Glossary | CSRC*. EN-US. URL: https://csrc.nist.gov/glossary/term/confidentiality (visited on 05/20/2022).

[52] CSRC Content Editor. *integrity - Glossary | CSRC*. EN-US. URL: https://csrc.nist.gov/glossary/term/integrity (visited on 05/20/2022).

[53] CSRC Content Editor. *privacy - Glossary | CSRC*. EN-US. URL: https://csrc.nist.gov/glossary/term/privacy (visited on 05/20/2022).

[54] Charles Feng, Shuning Wu, and Ningwei Liu. “A user-centric machine learning framework for cyber security operations center”. In: *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. July 2017, pp. 173–175. DOI: 10.1109/ISI.2017.8004902.

[55] Gilberto Fernandes et al. “A comprehensive survey on network anomaly detection”. en. In: *Telecommunication Systems* 70.3 (Mar. 2019), pp. 447–489. ISSN: 1572-9451. DOI: 10.1007/s11235-018-0475-8. URL: https://doi.org/10.1007/s11235-018-0475-8 (visited on 04/17/2022).

[56] Mohamed Amine Ferrag et al. “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study”. en. In: *Journal of Information Security and Applications* 50 (Feb. 2020), p. 102419. ISSN: 2214-2126. DOI: 10.1016/j.jisa.2019.102419. URL: https://www.sciencedirect.com/science/article/pii/S2214212619305046 (visited on 05/22/2022).

[57] Erik M. Ferragut et al. “Automatic construction of anomaly detectors from graphical models”. In: *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*. Apr. 2011, pp. 9–16. DOI: 10.1109/CICYBS.2011.5949386.

[58] Sally Floyd, K. K. Ramakrishnan, and David L. Black. *The Addition of Explicit Congestion Notification (ECN) to IP*. Request for Comments RFC 3168. Num Pages: 63. Internet Engineering Task Force, Sept. 2001. DOI: 10.17487/RFC3168. URL: https://datatracker.ietf.org/doc/rfc3168 (visited on 03/09/2022).

[59] Ruth Fong and Andrea Vedaldi. “Explanations for Attributing Deep Neural Network Predictions”. en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 149–167. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_8. URL: https://doi.org/10.1007/978-3-030-28954-6_8 (visited on 04/02/2022).

[60] The Open Information Secuirty Foundation. *Suricata Front Page*. en-US. URL: https://suricata.io/ (visited on 03/08/2022).

[61] Patricia Fusch and Lawrence Ness. “Are We There Yet? Data Saturation in Qualitative Research”. In: *The Qualitative Report* 20.9 (Sept. 2015), pp. 1408–1416. ISSN: 1052-0147. DOI: 10.46743/2160-3715/2015.2281. URL: https://nsuworks.nova.edu/tqr/vol20/iss9/3.

[62] S. García et al. “An empirical comparison of botnet detection methods”. en. In: *Computers & Security* 45 (Sept. 2014), pp. 100–123. ISSN: 0167-4048. DOI: 10.1016/j.cose.2014.05.011. URL: https://www.sciencedirect.com/science/article/pii/S0167404814000923 (visited on 04/20/2022).

[63] P. García-Teodoro et al. “Anomaly-based network intrusion detection: Techniques, systems and challenges”. In: *Computers & Security* 28.1 (2009), pp. 18–28. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2008.08.003. URL: https://www.sciencedirect.com/science/article/pii/S0167404808000692.

[64] Gartner, Inc. *Gartner's SOAR Market Guide | Orchestration & Automation*. en-US. URL: https://d3security.com/resources/gartner-soar-market-guide/ (visited on 04/17/2022).

[65] Prashant Gohel, Priyanka Singh, and Manoranjan Mohanty. “Explainable AI: current status and future directions”. In: *arXiv:2107.07045 [cs]* (July 2021). arXiv: 2107.07045. URL: http://arxiv.org/abs/2107.07045 (visited on 01/19/2022).

[66] John R. Goodall et al. “Situ: Identifying and Explaining Suspicious Behavior in Networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019). Conference Name: IEEE Transactions on Visualization and Computer Graphics, pp. 204–214. ISSN: 1941-0506. DOI: 10.1109/TVCG.2018.2865029.

[67] Google. *HTTPS encryption on the web – Google Transparency Report*. URL: https://transparencyreport.google.com/https/overview?hl=en (visited on 04/24/2022).

[68] Google Brain Team. *TensorFlow*. en. URL: https://www.tensorflow.org/ (visited on 03/03/2022).

[69] Google Brain Team. *Train a Keras model — fit*. URL: https://keras.rstudio.com/reference/fit.html (visited on 03/08/2022).

[70] Gordon Lyon. *Nmap: the Network Mapper - Free Security Scanner*. URL: https://nmap.org/ (visited on 03/08/2022).

[71] David Gunning et al. “XAI—Explainable artificial intelligence”. EN. In: *Science Robotics* (Dec. 2019). Publisher: American Association for the Advancement of Science. DOI: 10.1126/scirobotics.aay7120. URL: https://www.science.org/doi/abs/10.1126/scirobotics.aay7120 (visited on 01/19/2022).

[72] Alka Gupta and Lalit Sen Sharma. “Performance Evaluation of Snort and Suricata Intrusion Detection Systems on Ubuntu Server”. In: *Proceedings of ICRIC 2019*. Ed. by Pradeep Kumar Singh et al. Cham: Springer International Publishing, 2020, pp. 811–821. ISBN: 978-3-030-29407-6.

[73] Karthik S. Gurumoorthy et al. *Efficient Data Representation by Selecting Prototypes with Importance Weights*. Tech. rep. arXiv:1707.01212. arXiv:1707.01212 [cs, stat] type: article. arXiv, Aug. 2019. DOI: 10.48550/arXiv.1707.01212. URL: http://arxiv.org/abs/1707.01212 (visited on 05/14/2022).

[74] W. Haider et al. “Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling”. In: *Journal of Network and Computer Applications* 87 (2017), pp. 185–192. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2017.03.018. URL: https://www.sciencedirect.com/science/article/pii/S1084804517301273.

[75] Yasir Hamid et al. “Benchmark Datasets for Network Intrusion Detection: A Review”. In: *International Journal of Network Security* (Jan. 2018). DOI: 10.6633/IJNS.2018xx.20(x).xx).

[76] Balázs Péter Hámornik and Csaba Krasznay. “A Team-Level Perspective of Human Factors in Cyber Security: Security Operations Centers”. en. In: *Advances in Human Factors in Cybersecurity*. Ed. by Denise Nicholson. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2018, pp. 224–236. ISBN: 978-3-319-60585-2. DOI: 10.1007/978-3-319-60585-2_21.

[77] Kazuyuki Hara, Daisuke Saito, and Hayaru Shouno. “Analysis of function of rectified linear unit used in deep learning”. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. 2015, pp. 1–8. DOI: 10.1109/IJCNN.2015.7280578.

[78] Jeff Heaton. *Introduction to Neural Networks with Java*. en. Google-Books-ID: Swlcw7M4uD8C. Heaton Research, Inc., 2008. ISBN: 978-1-60439-008-7.

[79] Thomas Hellström, Virginia Dignum, and Suna Bensch. *Bias in Machine Learning – What is it Good for?* Tech. rep. arXiv:2004.00686. arXiv:2004.00686 [cs] type: article. arXiv, Sept. 2020. DOI: 10.48550/arXiv.2004.00686. URL: http://arxiv.org/abs/2004.00686 (visited on 05/22/2022).

[80] Xuan Dau Hoang and Quynh Chi Nguyen. “Botnet Detection Based On Machine Learning Techniques Using DNS Query Data”. en. In: *Future Internet* 10.5 (May 2018). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 43. ISSN: 1999-5903. DOI: 10.3390/fi10050043. URL: https://www.mdpi.com/1999-5903/10/5/43 (visited on 02/16/2022).

[81] Rick Hofstede et al. “Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX”. In: *IEEE Communications Surveys & Tutorials* 16.4 (2014), pp. 2037–2064. ISSN: 1553-877X. DOI: 10.1109/COMST.2014.2321898. URL: https://ieeexplore.ieee.org/document/6814316 (visited on 05/13/2022).

[82] Seunghoon Hong et al. “Interpretable Text-to-Image Synthesis with Hierarchical Semantic Layout Generation”. en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 77–95. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_5. URL: https://doi.org/10.1007/978-3-030-28954-6_5 (visited on 04/02/2022).

[83] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. en. In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366. ISSN: 08936080. DOI: 10.1016/0893-6080(89)90020-8. URL: https://linkinghub.elsevier.com/retrieve/pii/0893608089900208 (visited on 05/25/2022).

[84] Weihua Hu et al. “Unsupervised Discrete Representation Learning”. en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 97–119. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_6. URL: https://doi.org/10.1007/978-3-030-28954-6_6 (visited on 04/02/2022).

[85] IBM IBM. *Cost of a Data Breach Report 2020 | IBM*. en. 2020. URL: https://www.ibm.com/security/digital-assets/cost-data-breach-report/ (visited on 05/12/2021).

[86] Information Sciences Institute University of Southern California. *Transmission Control Protocol*. Request for Comments RFC 793. Num Pages: 91. Internet Engineering Task Force, Sept. 1981. DOI: 10.17487/RFC0793. URL: https://datatracker.ietf.org/doc/rfc793 (visited on 03/09/2022).

[87] Muhammad Usama Islam et al. “The Past, Present, and Prospective Future of XAI: A Comprehensive Review”. en. In: *Explainable Artificial Intelligence for Cyber Security: Next Generation Artificial Intelligence*. Ed. by Mohiuddin Ahmed et al. Studies in Computational Intelligence. Cham: Springer International Publishing, 2022, pp. 1–29. ISBN: 978-3-030-96630-0. DOI: 10.1007/978-3-030-96630-0_1. URL: https://doi.org/10.1007/978-3-030-96630-0_1 (visited on 05/19/2022).

[88] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. “Perturbation-based methods for explaining deep neural networks: A survey”. en. In: *Pattern Recognition Letters* 150 (Oct. 2021), pp. 228–234. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2021.06.030. URL: https://www.sciencedirect.com/science/article/pii/S0167865521002440 (visited on 05/23/2022).

[89] Parth Jain. “Machine Learning versus Deep Learning for Malware Detection”. en. Master of Science. San Jose, CA, USA: San Jose State University, May 2019. DOI: 10.31979/etd.56y7-b74e. URL: https://scholarworks.sjsu.edu/etd_projects/704 (visited on 05/22/2022).

[90] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. “Feature relevance quantification in explainable AI: A causal problem”. In: *arXiv:1910.13413 [cs, stat]* (Nov. 2019). arXiv: 1910.13413. URL: http://arxiv.org/abs/1910.13413 (visited on 04/13/2022).

[91] Jerome H. Friedman. “Greedy function approximation: A gradient boosting machine.” In: *The Annals of Statistics* 29.5 (Oct. 2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

[92] M. I. Jordan and T. M. Mitchell. “Machine learning: Trends, perspectives, and prospects”. EN. In: *Science* (July 2015). Publisher: American Association for the Advancement of Science. DOI: 10.1126/science.aaa8415. URL: https://www.science.org/doi/abs/10.1126/science.aaa8415 (visited on 01/18/2022).

[93] Josiah Dykstra. *1. Introduction to Cybersecurity Science - Essential Cybersecurity Science [Book]*. en. ISBN: 9781491920947. Feb. 2022. URL: https://www.oreilly.com/library/view/essential-cybersecurity-science/9781491921050/ch01.html (visited on 02/16/2022).

[94] Louise H. Kidder and Charles M. Judd. *Research Methods in Social Relations, Fifth Edition*. Fifth Edition, Highlighting. Holt, Rinehart and Winston, Jan. 1986.

[95] Rupesh Raj Karn et al. "Cryptomining Detection in Container Clouds Using System Calls and Explainable Machine Learning". In: *IEEE Transactions on Parallel and Distributed Systems* 32.3 (Mar. 2021). Conference Name: IEEE Transactions on Parallel and Distributed Systems, pp. 674–691. ISSN: 1558-2183. DOI: 10.1109/TPDS.2020.3029088.

[96] Keras. *Keras: the Python deep learning API*. URL: https://keras.io/ (visited on 03/03/2022).

[97] Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. "Examples are not enough, learn to criticize! Criticism for interpretability". English (US). In: *Advances in Neural Information Processing Systems* (Jan. 2016). 30th Annual Conference on Neural Information Processing Systems, NIPS 2016 ; Conference date: 05-12-2016 Through 10-12-2016, pp. 2288–2296. ISSN: 1049-5258.

[98] Gulshan Kumar, Krishan Kumar, and Monika Sachdeva. "The use of artificial intelligence based techniques for intrusion detection: a review". In: *Artificial Intelligence Review* 34.4 (2010), pp. 369–387. ISSN: 1573-7462. DOI: 10.1007/s10462-010-9179-5. URL: https://doi.org/10.1007/s10462-010-9179-5.

[99] Manish Kumar, M. Hanumanthappa, and T. V. Suresh Kumar. "Encrypted Traffic and IPsec Challenges for Intrusion Detection System". en. In: *Proceedings of International Conference on Advances in Computing*. Ed. by Aswatha Kumar M., Selvarani R., and T V Suresh Kumar. Advances in Intelligent Systems and Computing. New Delhi: Springer India, 2012, pp. 721–727. ISBN: 978-81-322-0740-5. DOI: 10.1007/978-81-322-0740-5_86.

[100] Vinod Kumar. "Signature Based Intrusion Detection System Using SNORT". In: *International Journal of Computer Applications & Information Technology* 1 (Nov. 2012), p. 7.

[101] Roshan Kumari and Saurabh Srivastava. "Machine Learning: A Review on Binary Classification". In: *International Journal of Computer Applications* 160 (Feb. 2017), pp. 11–15. DOI: 10.5120/ijca2017913083.

[102] Jordan Lam and Robert Abbas. "Machine Learning based Anomaly Detection for 5G Networks". In: *arXiv:2003.03474 [cs, stat]* (Mar. 2020). arXiv: 2003.03474. URL: http://arxiv.org/abs/2003.03474 (visited on 04/24/2022).

[103] Jian-hua Li. “Cyber security meets artificial intelligence: a survey”. en. In: *Frontiers of Information Technology & Electronic Engineering* 19.12 (Dec. 2018), pp. 1462–1474. ISSN: 2095-9230. DOI: 10.1631/FITEE.1800573. URL: https://doi.org/10.1631/FITEE.1800573 (visited on 05/14/2022).

[104] Kai Liu et al. *A review of knowledge graph application scenarios in cyber security*. Tech. rep. arXiv:2204.04769. arXiv:2204.04769 [cs] type: article. arXiv, Apr. 2022. DOI: 10.48550/arXiv.2204.04769. URL: http://arxiv.org/abs/2204.04769 (visited on 05/14/2022).

[105] Jörn Lötsch, Dario Kringel, and Alfred Ultsch. “Explainable Artificial Intelligence (XAI) in Biomedicine: Making AI Decisions Trustworthy for Physicians and Patients”. en. In: *BioMedInformatics* 2.1 (Mar. 2022). Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, pp. 1–17. DOI: 10.3390/biomedinformatics2010001. URL: https://www.mdpi.com/2673-7426/2/1/1 (visited on 01/19/2022).

[106] Xianguang Lu, Xuehui Du, and Wenjuan Wang. “An Alert Aggregation Algorithm Based on K-means and Genetic Algorithm”. In: *IOP Conference Series: Materials Science and Engineering* 435 (Nov. 2018). Publisher: IOP Publishing, p. 012031. DOI: 10.1088/1757-899x/435/1/012031. URL: https://doi.org/10.1088/1757-899x/435/1/012031.

[107] Michael J. de Lucia and Chase Cotton. “Detection of Encrypted Malicious Network Traffic using Machine Learning”. In: *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*. ISSN: 2155-7586. Nov. 2019, pp. 1–6. DOI: 10.1109/MILCOM47813.2019.9020856.

[108] Scott Lundberg. *shap2: A unified approach to explain the output of any machine learning model.* URL: https://pypi.org/project/shap2/ (visited on 05/02/2022).

[109] Scott Lundberg. *slundberg/shap*. original-date: 2016-11-22T19:17:08Z. May 2022. URL: https://github.com/slundberg/shap (visited on 05/02/2022).

[110] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[111] M. Bagnulo et al. *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*. Apr. 2011. URL: https://www.ietf.org/rfc/rfc6146.txt (visited on 02/25/2022).

[112] Zhuo Ma et al. “A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms”. In: *IEEE Access* 7 (2019). Conference Name: IEEE Access, pp. 21235–21245. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2896003.

[113] Gabriel Maciá-Fernández et al. "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs". In: *Computers & Security* 73 (2018), pp. 411–424. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2017.11.004. URL: https://www.sciencedirect.com/science/article/pii/S0167404817302353.

[114] M. Majid and K. Ariffi. "Success Factors for Cyber Security Operation Center (SOC) Establishment". In: Oct. 2019. ISBN: 978-1-63190-198-0. URL: https://eudl.eu/doi/10.4108/eai.18-7-2019.2287841 (visited on 01/13/2022).

[115] Shraddha Mane and Dattaraj Rao. "Explaining Network Intrusion Detection System Using Explainable AI Framework". In: *arXiv:2103.07110 [cs]* (Mar. 2021). arXiv: 2103.07110. URL: http://arxiv.org/abs/2103.07110 (visited on 04/22/2022).

[116] Manu Joseph. *Interpretability part 3: opening the black box with LIME and SHAP*. en-US. Section: 2019 Dec Tutorials, Overviews. URL: https://www.kdnuggets.com/interpretability-part-3-lime-and-shap.html/ (visited on 05/25/2022).

[117] Stephen Marsland. *Machine Learning: An Algorithmic Perspective*. en. 2nd ed. Chapman and Hall/CRC, Oct. 2014. ISBN: 978-0-429-10250-9. DOI: 10.1201/b17476. URL: https://www.taylorfrancis.com/books/9781466583337 (visited on 03/04/2022).

[118] Sherin Mary Mathews. "Explainable Artificial Intelligence Applications in NLP, Biomedical, and Malware Classification: A Literature Review". en. In: *Intelligent Computing*. Ed. by Kohei Arai, Rahul Bhatia, and Supriya Kapoor. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2019, pp. 1269–1292. ISBN: 978-3-030-22868-2. DOI: 10.1007/978-3-030-22868-2_90.

[119] Ninareh Mehrabi et al. "A Survey on Bias and Fairness in Machine Learning". In: *ACM Computing Surveys* 54.6 (July 2021), 115:1–115:35. ISSN: 0360-0300. DOI: 10.1145/3457607. URL: https://doi.org/10.1145/3457607 (visited on 05/22/2022).

[120] Micro Focus. *2020 State of Security Operations | Analyst Report*. en. 2020. URL: https://www.microfocus.com/en-us/assets/cyberres/2020-state-of-security-operations (visited on 04/17/2022).

[121] Microsoft. *Understanding Remote Desktop Protocol (RDP) - Windows Server*. en-us. URL: https://docs.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol (visited on 05/23/2022).

[122] Milena Pavlovic. "Interpretability and explainability in machine learning". en. In: *MLS research seminar at University of Oslo IFI, 5th of June 2020* (June 2020), p. 13. URL: https://www.mn.uio.no/ifi/english/about/organisation/mls/seminar/200228/hein.pdf.

[123] Matthew B. Miles and A. Michael Huberman. *Qualitative data analysis: An expanded sourcebook, 2nd ed.* Qualitative data analysis: An expanded sourcebook, 2nd ed. Pages: xiv, 338. Thousand Oaks, CA, US: Sage Publications, Inc, 1994. ISBN: 0-8039-4653-8 (Hardcover); 0-8039-5540-5 (Paperback).

[124] Natalia Miloslavskaya. "Analysis of SIEM Systems and Their Usage in Security Operations and Security Intelligence Centers". en. In: *Biologically Inspired Cognitive Architectures (BICA) for Young Scientists*. Ed. by Alexei V. Samsonovich and Valentin V. Klimov. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2018, pp. 282–288. ISBN: 978-3-319-63940-6. DOI: 10.1007/978-3-319-63940-6_40.

[125] Christoph Molnar. *Interpretable Machine Learning*. URL: https://christophm.github.io/interpretable-ml-book/index.html#summary (visited on 04/02/2022).

[126] Grégoire Montavon et al. "Layer-Wise Relevance Propagation: An Overview". en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 193–209. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_10. URL: https://doi.org/10.1007/978-3-030-28954-6_10 (visited on 04/02/2022).

[127] Sonya J. Morgan et al. "Case Study Observational Research: A Framework for Conducting Case Study Research Where Observation Data Are the Focus". eng. In: *Qualitative Health Research* 27.7 (June 2017), pp. 1060–1068. ISSN: 1049-7323. DOI: 10.1177/1049732316649160.

[128] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. "Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. arXiv:1905.07697 [cs, stat]. Jan. 2020, pp. 607–617. DOI: 10.1145/3351095.3372850. URL: http://arxiv.org/abs/1905.07697 (visited on 05/14/2022).

[129] Nour Moustafa and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set". In: (Jan. 2016), pp. 1–14. DOI: 10.1080/19393555.2015.1125974.

[130] Nour Moustafa and Jill Slay. "The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems". In: Nov. 2015, pp. 25–31. DOI: 10.1109/BADGERS.2015.014.

[131] Nour Moustafa and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". In: *2015 Military Communications and Information Systems Conference (MilCIS)*. 2015, pp. 1–6. DOI: 10.1109/MilCIS.2015.7348942.

[132] David Nathans. "Chapter 1 - Efficient operations: Building an operations center from the ground up". en. In: *Designing and Building Security Operations Center*. Ed. by David Nathans. Syngress, Jan. 2015, pp. 1–24. ISBN: 978-0-12-800899-7. DOI: 10.1016/B978-0-12-800899-7.00001-X. URL: https://www.sciencedirect.com/science/article/pii/B978012800899700001X (visited on 05/21/2022).

[133] Anh Nguyen, Jason Yosinski, and Jeff Clune. "Understanding Neural Networks via Feature Visualization: A Survey". en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 55–76. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_4. URL: https://doi.org/10.1007/978-3-030-28954-6_4 (visited on 04/02/2022).

[134] Robert Nisbet, Gary Miner, and Ken Yale. "Chapter 9 - Classification". In: *Handbook of Statistical Analysis and Data Mining Applications (Second Edition)*. Ed. by Robert Nisbet, Gary Miner, and Ken Yale. Second Edition. Boston: Academic Press, 2018, pp. 169–186. ISBN: 978-0-12-416632-5. DOI: https://doi.org/10.1016/B978-0-12-416632-5.00009-8. URL: https://www.sciencedirect.com/science/article/pii/B9780124166325000098.

[135] Helen Noble and Joanna Smith. "Issues of validity in qualitative research". In: *Evidence-based nursing* 18 (Feb. 2015). DOI: 10.1136/eb-2015-102054.

[136] Megan M. Nyre-Yu. "Determining System Requirements for Human-Machine Integration in Cyber Security Incident Response". en. thesis. Purdue University Graduate School, Oct. 2019. DOI: 10.25394/PGS.10014803.v1. URL: https://hammer.purdue.edu/articles/thesis/Determining_System_Requirements_for_Human-Machine_Integration_in_Cyber_Security_Incident_Response/10014803/1 (visited on 05/26/2022).

[137] Sean Oesch et al. "An Assessment of the Usability of Machine Learning Based Tools for the Security Operations Center". In: *arXiv:2012.09013 [cs]* (Dec. 2020). arXiv: 2012.09013. URL: http://arxiv.org/abs/2012.09013 (visited on 04/23/2022).

[138] Seong Joon Oh, Bernt Schiele, and Mario Fritz. "Towards Reverse-Engineering Black-Box Neural Networks". en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Cham: Springer International Publishing, 2019, pp. 121–144. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_7. URL: https://doi.org/10.1007/978-3-030-28954-6_7 (visited on 04/02/2022).

[139] Morufu Olalere et al. "Proposed Discriminative Lexical Features for Real-time Detection of Malware Uniform Resource Locator". In: *Indian Journal of Science and Technology* 9 (Dec. 2016). DOI: 10.17485/ijst/2016/v9i46/107081.

[140] Cyril Onwubiko. "Cyber security operations centre: Security monitoring for protecting business and supporting cyber defense strategy". In: *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*. June 2015, pp. 1–10. DOI: 10.1109/CyberSA.2015.7166125.

[141] Long Ouyang et al. *Training language models to follow instructions with human feedback*. Tech. rep. arXiv:2203.02155. arXiv:2203.02155 [cs] type: article. arXiv, Mar. 2022. DOI: 10.48550/arXiv.2203.02155. URL: http://arxiv.org/abs/2203.02155 (visited on 05/14/2022).

[142] Palo Alto Networks. *Command and Control Explained*. en-US. Feb. 2022. URL: https://www.paloaltonetworks.com/cyberpedia/command-and-control-explained (visited on 02/16/2022).

[143] Jose N. Paredes et al. *On the Importance of Domain-specific Explanations in AI-based Cybersecurity Systems (Technical Report)*. Tech. rep. arXiv:2108.02006. arXiv:2108.02006 [cs] type: article. arXiv, Aug. 2021. DOI: 10.48550/arXiv.2108.02006. URL: http://arxiv.org/abs/2108.02006 (visited on 05/19/2022).

[144] Manoranjan Pradhan et al. *Intrusion Detection System (IDS) and Their Types*. English. Chap. Archive Location: intrusion-detection-system-ids-and-their-types ISBN: 9781466687615 Publisher: IGI Global. Jan. 2001. URL: https://www.igi-global.com/gateway/chapter/www.igi-global.com/gateway/chapter/143973 (visited on 05/22/2022).

[145] Pritha Bhandari. *A step-by-step guide to data collection*. en-US. June 2020. URL: https://www.scribbr.com/methodology/data-collection/ (visited on 05/01/2022).

[146] Pritha Bhandari. *An introduction to qualitative research*. en-US. June 2020. URL: https://www.scribbr.com/methodology/qualitative-research/ (visited on 05/01/2022).

[147] Python community. *PyPI · The Python Package Index*. en. URL: https://pypi.org/ (visited on 05/01/2022).

[148] J. R. Quinlan. "Induction of decision trees". In: *Machine Learning* 1.1 (Mar. 1986), pp. 81–106. ISSN: 1573-0565. DOI: 10.1007/BF00116251. URL: https://doi.org/10.1007/BF00116251.

[149] Daan Raman et al. "DNS Tunneling for Network Penetration". In: *Information Security and Cryptology – ICISC 2012*. Ed. by Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 65–77. ISBN: 978-3-642-37682-5.

[150] Megan L. Ranney et al. "Interview-Based Qualitative Research in Emergency Care Part II: Data Collection, Analysis and Results Reporting". In: *Academic emergency medicine : official journal of the Society for Academic Emergency Medicine* 22.9 (Sept. 2015), pp. 1103–1112. ISSN: 1069-6563. DOI: 10.1111/acem.12735. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4560670/ (visited on 05/24/2022).

[151] Routhu Srinivasa Rao and Alwyn Roshan Pais. “Detection of phishing websites using an efficient feature-based machine learning framework”. en. In: *Neural Computing and Applications* 31.8 (Aug. 2019), pp. 3851–3873. ISSN: 1433-3058. DOI: 10.1007/s00521-017-3305-0. URL: https://doi.org/10.1007/s00521-017-3305-0 (visited on 05/22/2022).

[152] A. Rawal et al. “Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges and Perspectives”. In: *IEEE Transactions on Artificial Intelligence* 1.01 (Dec. 5555). Place: Los Alamitos, CA, USA Publisher: IEEE Computer Society, pp. 1–1. DOI: 10.1109/TAI.2021.3133846.

[153] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, pp. 1135–1144.

[154] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Model-Agnostic Interpretability of Machine Learning”. In: *arXiv:1606.05386 [cs, stat]* (June 2016). arXiv: 1606.05386. URL: http://arxiv.org/abs/1606.05386 (visited on 04/01/2022).

[155] Marco Tulio Correia Ribeiro. *lime*. original-date: 2016-03-15T22:18:10Z. May 2022. URL: https://github.com/marcotcr/lime (visited on 05/02/2022).

[156] Markus Ring et al. “A survey of network-based intrusion detection data sets”. In: *Computers & Security* 86 (2019), pp. 147–167. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2019.06.005. URL: https://www.sciencedirect.com/science/article/pii/S016740481930118X.

[157] Markus Ring et al. “Flow-Based Benchmark Data Sets for Intrusion Detection”. In: June 2017.

[158] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre de Montjoye. “Estimating the success of re-identifications in incomplete datasets using generative models”. en. In: *Nature Communications* 10.1 (July 2019). Number: 1 Publisher: Nature Publishing Group, p. 3069. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10933-3. URL: https://www.nature.com/articles/s41467-019-10933-3 (visited on 05/22/2022).

[159] Eirik Rossen. *DNS*. nb. May 2021. URL: http://snl.no/DNS (visited on 02/15/2022).

[160] Olga Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. Tech. rep. arXiv:1409.0575. arXiv:1409.0575 [cs] type: article. arXiv, Jan. 2015. URL: http://arxiv.org/abs/1409.0575 (visited on 05/22/2022).

[161] Julie Ryan. *Leading Issues in Information Warfare and Security Research*. en. Google-Books-ID: oukNfumrXpcC. Academic Conferences Limited, 2011. ISBN: 978-1-908272-08-9.

[162] Sherif Saad, William Briguglio, and Haytham Elmiligi. *The Curious Case of Machine Learning In Malware Detection*. Tech. rep. arXiv:1905.07573. arXiv:1905.07573 [cs] type: article. arXiv, May 2019. DOI: 10.48550/arXiv.1905.07573. URL: http://arxiv.org/abs/1905.07573 (visited on 05/22/2022).

[163] Ozgur Koray Sahingoz et al. "Machine learning based phishing detection from URLs". en. In: *Expert Systems with Applications* 117 (Mar. 2019), pp. 345–357. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2018.09.029. URL: https://www.sciencedirect.com/science/article/pii/S0957417418306067 (visited on 05/22/2022).

[164] Fatima Salahdine, Zakaria El Mrabet, and Naima Kaabouch. "Phishing Attacks Detection – A Machine Learning-Based Approach". In: *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. arXiv:2201.10752 [cs]. Dec. 2021, pp. 0250–0255. DOI: 10.1109/UEMCON53757.2021.9666627. URL: http://arxiv.org/abs/2201.10752 (visited on 05/22/2022).

[165] Wojciech Samek et al. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning | SpringerLink*. 2019. URL: https://link.springer.com/book/10.1007/978-3-030-28954-6#editorsandaffiliations (visited on 04/02/2022).

[166] Spyridon Samonas and David Coss. "The CIA strikes back: Redefining confidentiality, integrity and availability in security." In: *Journal of Information System Security* 10.3 (2014).

[167] George AF Seber and Alan J Lee. *Linear regression analysis*. Vol. 329. John Wiley & Sons, 2012.

[168] M. Shanker, M. Y. Hu, and M. S. Hung. "Effect of data standardization on neural network training". en. In: *Omega* 24.4 (Aug. 1996), pp. 385–397. ISSN: 0305-0483. DOI: 10.1016/0305-0483(96)00010-2. URL: https://www.sciencedirect.com/science/article/pii/0305048396000102 (visited on 05/01/2022).

[169] Lloyd S. Shapley. *A Value for N-Person Games*. Santa Monica, CA: RAND Corporation, 1952. DOI: 10.7249/P0295.

[170] Iman Sharafaldin., Arash Habibi Lashkari., and Ali A. Ghorbani. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization". In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP,* INSTICC. SciTePress, 2018, pp. 108–116. ISBN: 978-989-758-282-0. DOI: 10.5220/0006639801080116.

[171] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *arXiv:1704.02685 [cs]* (Oct. 2019). arXiv: 1704.02685. URL: http://arxiv.org/abs/1704.02685 (visited on 04/12/2022).

[172] João Vitor V. Silva et al. “A statistical analysis of intrinsic bias of network security datasets for training machine learning mechanisms”. en. In: *Annals of Telecommunications* (Feb. 2022). ISSN: 1958-9395. DOI: 10.1007/s12243-021-00904-5. URL: https://doi.org/10.1007/s12243-021-00904-5 (visited on 05/22/2022).

[173] David Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. EN. In: *Science* (Dec. 2018). Publisher: American Association for the Advancement of Science. DOI: 10.1126/science.aar6404. URL: https://www.science.org/doi/abs/10.1126/science.aar6404 (visited on 01/13/2022).

[174] Jay Sinha. *Efficient-CNN-BiLSTM-for-Network-IDS*. original-date: 2021-01-03T19:13:48Z. May 2022. URL: https://github.com/razor08/Efficient-CNN-BiLSTM-for-Network-IDS (visited on 05/25/2022).

[175] Jay Sinha and M. Manollas. “Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection”. In: *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*. AIPR 2020. New York, NY, USA: Association for Computing Machinery, June 2020, pp. 223–231. ISBN: 978-1-4503-7551-1. DOI: 10.1145/3430199.3430224. URL: https://doi.org/10.1145/3430199.3430224 (visited on 05/25/2022).

[176] Dylan Slack et al. “Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods”. In: *arXiv:1911.02508 [cs, stat]* (Feb. 2020). arXiv: 1911.02508. URL: http://arxiv.org/abs/1911.02508 (visited on 04/13/2022).

[177] Michael R. Smith et al. *Mind the Gap: On Bridging the Semantic Gap between Machine Learning and Information Security*. Tech. rep. arXiv:2005.01800. arXiv:2005.01800 [cs, stat] type: article. arXiv, May 2020. DOI: 10.48550/arXiv.2005.01800. URL: http://arxiv.org/abs/2005.01800 (visited on 05/18/2022).

[178] Robin Sommer and Vern Paxson. “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection”. In: *2010 IEEE Symposium on Security and Privacy*. ISSN: 2375-1207. May 2010, pp. 305–316. DOI: 10.1109/SP.2010.25.

[179] Jonathan M. Spring et al. *Machine Learning in Cybersecurity: A Guide*. en. Tech. rep. Section: Technical Reports. Carnegie Mellon University Software Engineering Institute Pittsburgh United States, Jan. 2019. URL: https://apps.dtic.mil/sti/citations/AD1082647 (visited on 05/22/2022).

[180] Farhana Sultana. “Reflexivity”. In: Mar. 2017, pp. 1–5. DOI: 10.1002/9781118786352.wbieg0686.

[181] I Sumaiya Thaseen, B Poorva, and P Sai Ushasree. “Network Intrusion Detection using Machine Learning Techniques”. In: *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. Feb. 2020, pp. 1–7. DOI: 10.1109/ic-ETITE47903.2020.148.

[182] Sathya Chandran Sundaramurthy et al. “A Tale of Three Security Operation Centers”. In: *Proceedings of the 2014 ACM Workshop on Security Information Workers*. SIW ’14. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 43–50. ISBN: 978-1-4503-3152-4. DOI: 10.1145/2663887.2663904. URL: https://doi.org/10.1145/2663887.2663904 (visited on 01/13/2022).

[183] Mukund Sundararajan and Amir Najmi. “The many Shapley values for model explanation”. In: *arXiv:1908.08474 [cs, econ]* (Feb. 2020). arXiv: 1908.08474. URL: http://arxiv.org/abs/1908.08474 (visited on 04/13/2022).

[184] Hatma Suryotrisongko et al. “Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing”. In: *IEEE Access* 10 (2022). Conference Name: IEEE Access, pp. 34613–34624. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3162588.

[185] Arief Rama Syarif and Windu Gata. “Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm”. In: *2017 11th International Conference on Information Communication Technology and System (ICTS)*. 2017, pp. 181–186. DOI: 10.1109/ICTS.2017.8265667.

[186] Keras Team. *Keras documentation: Dense layer*. en. URL: https://keras.io/api/layers/core_layers/dense/ (visited on 03/04/2022).

[187] Keras Team. *Keras documentation: Dropout layer*. en. URL: https://keras.io/api/layers/regularization_layers/dropout/ (visited on 03/04/2022).

[188] Gabriel Terejanu et al. *Explainable Deep Modeling of Tabular Data using TableGraphNet*. Tech. rep. arXiv:2002.05205. arXiv:2002.05205 [cs] type: article. arXiv, Feb. 2020. DOI: 10.48550/arXiv.2002.05205. URL: http://arxiv.org/abs/2002.05205 (visited on 05/14/2022).

[189] The MITRE corporation. *Apache Log4j : List of security vulnerabilities*. 2022. URL: https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-37215/Apache-Log4j.html (visited on 05/10/2022).

[190] The Open Information Secuirty Foundation. *6. Suricata Rules — Suricata 6.0.4 documentation*. URL: https://suricata.readthedocs.io/en/suricata-6.0.4/rules/index.html (visited on 03/09/2022).

[191] U.S. Department of Homeland Security. *CWE - Common Weakness Enumeration*. URL: http://cwe.mitre.org/index.html (visited on 05/25/2022).

[192] Manfred Vielberth et al. “Security Operations Center: A Systematic Study and Open Challenges”. In: *IEEE Access* 8 (2020). Conference Name: IEEE Access, pp. 227756–227779. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3045514.

[193] Rahul Vigneswaran. *Intrusion Detection Systems*. original-date: 2018-09-22T09:24:43Z. May 2022. URL: https://github.com/rahulvigneswaran/Intrusion-Detection-Systems (visited on 05/25/2022).

[194] VMRay. *Indicators of Compromise (IOCs) and Artifacts: What's the Difference?* en-US. June 2020. URL: https://www.vmray.com/cyber-security-blog/indicators-of-compromise-artifacts-whats-the-difference/ (visited on 02/16/2022).

[195] Tuan Phan Vuong et al. "Decision tree-based detection of denial of service and command injection attacks on robotic vehicles". In: *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2015, pp. 1–6. DOI: 10.1109/WIFS.2015.7368559.

[196] Maonan Wang et al. "An Explainable Machine Learning Framework for Intrusion Detection Systems". In: *IEEE Access* 8 (2020). Conference Name: IEEE Access, pp. 73127–73141. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2988359.

[197] Łukasz Wawrowski et al. "Detecting anomalies and attacks in network traffic monitoring with classification methods and XAI-based explainability". en. In: *Procedia Computer Science*. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021 192 (Jan. 2021), pp. 2259–2268. ISSN: 1877-0509. DOI: 10.1016/j.procs.2021.08.239. URL: https://www.sciencedirect.com/science/article/pii/S1877050921017361 (visited on 04/20/2022).

[198] Patricia A H Williams and Vincent McCauley. "Always connected: The security challenges of the healthcare Internet of Things". In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. Dec. 2016, pp. 30–35. DOI: 10.1109/WF-IoT.2016.7845455.

[199] Jian Wu et al. "Citeseerx: Ai in a digital library search engine". In: *AI Magazine* 36.3 (2015), pp. 35–48.

[200] R.K. Yin. *Case Study Research and Applications: Design and Methods*. SAGE Publications, 2017. ISBN: 978-1-5063-3618-3. URL: https://books.google.no/books?id=6DwmDwAAQBAJ.

[201] Abdurrahim Yıldırım. *TCP 3-Way Handshake*. en. Jan. 2019. URL: https://medium.com/@yildirimabdrhm/tcp-3-way-handshake-2e4d4d674ff6 (visited on 03/09/2022).

[202] zscaler. *Encrypted Attacks Rise 314%*. en. Oct. 2021. URL: https://www.zscaler.com/blogs/security-research/encrypted-attacks-rise-314 (visited on 04/24/2022).

A Code

```python
#!/usr/bin/env python3

""" PRIMARY SOURCE
Primary source for code structure and influence:
Link: https://github.com/rahulvigneswaran/Intrusion-Detection-Systems
Author: Rahul Vigneswaran
Date: 23 Jun, 2020
"""

# The mostly cleaned up code for publication in the master thesis

# IMPORTS
#from turtle import back
import numpy as np
import pandas as pd
import argparse         # Haandter inputparsing

# Preprocessing
from sklearn.model_selection import StratifiedKFold
from imblearn.over_sampling import RandomOverSampler

# Metrics
from sklearn.metrics import accuracy_score

# Deep Neural Network
import tensorflow as tf
tf.compat.v1.disable_v2_behavior()
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation

# XAI - SHAP (SHapley Additive exPlanations)
import shap

# XAI - LIME
import lime
import lime.lime_tabular

# VARIABLER
    # UNSW .csv
data_unsw       = "/dataset/UNSW-NB15-CSV/"
nb_train        = "UNSW_NB15_training-set.csv"
nb_test         = "UNSW_NB15_testing-set.csv"

    # Analysis
analysis_folder = "analyse_data/"

    # DNN
dnn_model_save_dir = "dnn3layer_result/"

    # Global parsing values
skip_training = False
epochs = 6
skip_remove_attacks = False
removed_attacks_csv = analysis_folder+ \
        "unsw_only_recon_and_normal.csv"

def exit1(string=""):
    # Helper function when debugging and testing
    print("Calling exit: ",string)
    exit()

def train_dnn_model(model, traindata, trainlabel, x_test_2=None, y_test_2=None):
    global epochs

    # Change shape from (1000,193,1) -> (1000,193)
    traindata = traindata.reshape(len(traindata), len(traindata[0]))
    x_test_2 = x_test_2.reshape(len(x_test_2), len(x_test_2[0]))

    model.fit(traindata, trainlabel, batch_size=64, validation_data=(x_test_2,y_test_2), epochs=epochs)
    model.save(dnn_model_save_dir+"dnn_"+"sigmoid_model.hdf5")
    return model

def test_predictions(model, X, columns, normal_indx, attack_indx):
    pred = model.predict(X.reshape(X.shape[0], len(columns)))

    norm = pred[normal_indx]
    atkk = pred[attack_indx]
    print(f"Model prediction: normal/{norm}, attack/{atkk}")

    # Concept of how single predictions can look
    singel = X[240]
    single_pred = model.predict( np.array( [singel,].reshape(1,193) ) )
    print("Single prediction:", single_pred)

# Chooses one attack and one normal, calls xai_shap and xai_lime
def xai_common(model, X, y, columns, combined_data):
    """ combiend_data - Un-normalised data """

    # Find one normal and one attack prediction
    normal_indx = -1
    attack_indx = -1
    normal_indx_real = -1
    attack_indx_real = -1
    counter = 0

    # Nytt forsook paa aa finne ett angrep
    for index, label in y.items():
        if (label == 0):
            normal_indx = index
            normal_indx_real = counter
        else:
            attack_indx = index
            attack_indx_real = counter

        counter += 1

        if (normal_indx != -1) and (attack_indx != -1): break

    print("\nXAI_SHAP")
    xai_shap(model, X, y, columns, combined_data, normal_indx_real, attack_indx_real)
    print("\n\nXAI_LIME")
    xai_lime(model, X, X, columns, normal_indx_real, attack_indx_real)


def xai_shap(model, X, y, columns, combined_data, normal_indx, attack_indx):
    """ X: testdata, y: labels, combined_data: Raw data before one-hot
```

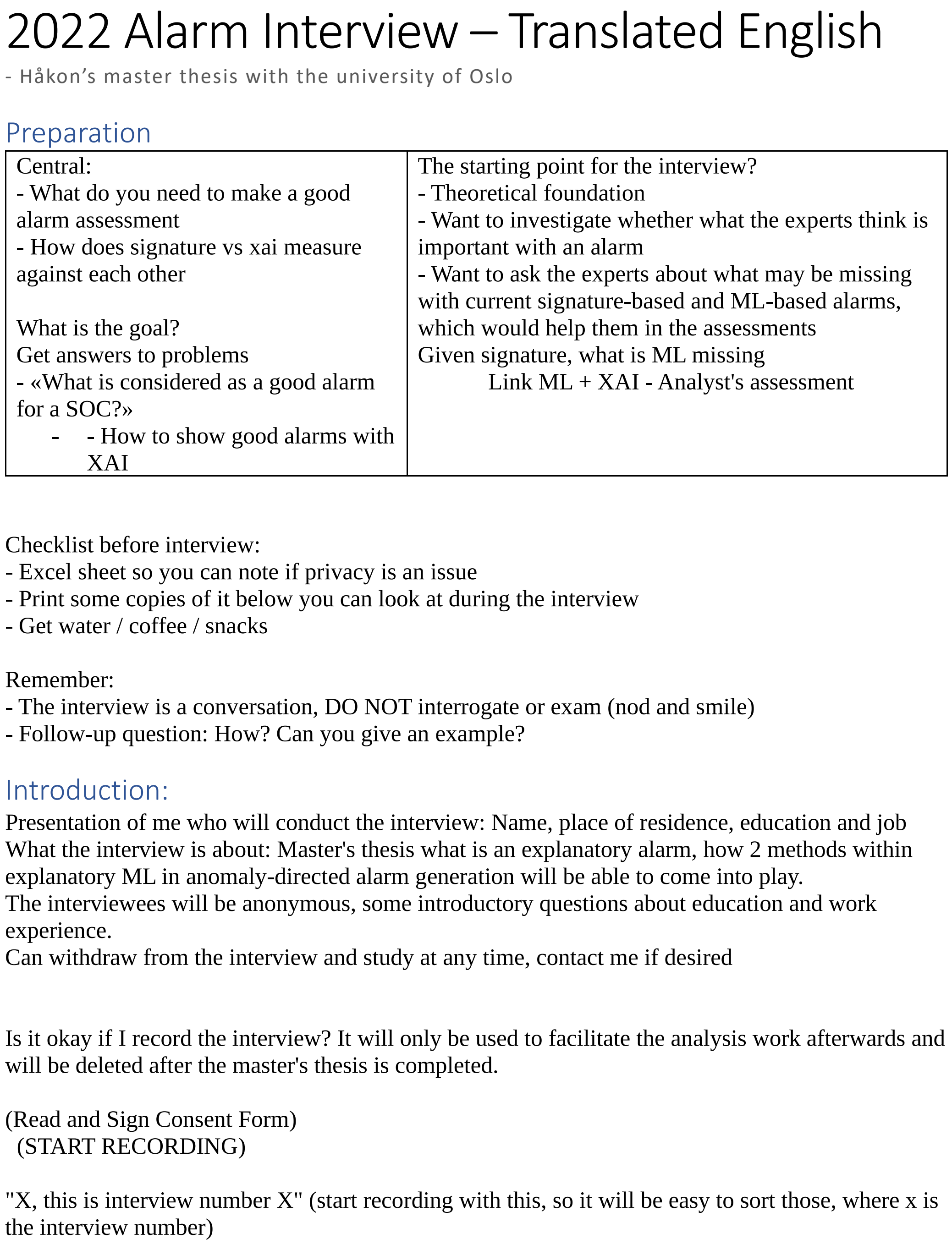
```
encoding and normalization"""

background = X[np.random.choice(X.shape[0], 1000, replace=False)].
reshape((1000,193))
e = shap.DeepExplainer(model, background)

# Create shap_values for one attack and one normal
shap_values_normal = (e.shap_values(X[normal_indx].reshape((1,193))
))[0][0]
shap_values_attack = (e.shap_values(X[attack_indx].reshape((1,193))
))[0][0]

# Sort the shap_values based on biggest impacts
shap_and_columns_normal = sorted(zip(columns, shap_values_normal),
key = lambda x: x[1])
shap_and_columns_attack = sorted(zip(columns, shap_values_attack),
key = lambda x: x[1])

from tabulate import tabulate

# For hver shap kolonne i top 5
# Hent verdien for attack_indx
# Hvis den ikke er i "un-preprocessed"/combined_data, sjekk "
preprocessed"/X
raw_row = combined_data.iloc[attack_indx]
shap_top5_attack = shap_and_columns_attack[-4:]

for i in range(len(shap_top5_attack)):
    colm_name = shap_top5_attack[i][0]      # eks: sttl, proto_ipv6
-no
    colm_val = raw_row[colm_name]           # eks: 255, 0
    shap_top5_attack[i] += (colm_val,)

    pretty_list = []                        # Used to print value,
numb_attack, numb_normal

    # For hver unike verdi i en kolonne (0, 1, 29, 255 osv..)
    # Tell antall for hver verdi som er et angrep eller normal
    for unique_column_values in np.unique(combined_data[colm_name])
:
        # Isolate the column with only one value (eks. 255)
        uniq_colms = combined_data[[colm_name, 'Class']]
        uniq_colms = uniq_colms.loc[uniq_colms[colm_name] ==
unique_column_values]
        uniq_values = uniq_colms.groupby([colm_name]).count()

        # For indexing, find index to column "Class"
        colm_class_indx = uniq_values.columns.get_loc('Class')

        # Sorter paa unike kolonner
        uniq_colms = uniq_colms.groupby(['Class']).count()
        # Index til kolonnen av interesse
        colm_class_indx = uniq_colms.columns.get_loc(colm_name)

        # Antall normal med verdi colm_val
        try:    numb_normal = uniq_colms.iloc[0, colm_class_indx]
        except: numb_normal = 0
        # Antall angrep med verdi colm_val
        try:    numb_attack = uniq_colms.iloc[1, colm_class_indx]
        except: numb_attack = 0

        # Check that number of normal and attack is equal the total
```

```
     expected
            assert (numb_normal + numb_attack == uniq_values.iloc[0,
    colm_class_indx])
            # Add the value and numbers to the list for printing
            pretty_list.append((unique_column_values, numb_normal,
    numb_attack))

        # Pretty print the columns values and the values number of
    normal and attack
        print(tabulate(pretty_list, headers=[colm_name+" value", "
    Normal", "Attack"]))

    print(tabulate(shap_top5_attack, headers=["Feature", "Score", "Flow
     Value"]))

def xai_lime(model, train, test, columns, normal_indx, attack_indx):
    """ normal_indx and attack_indx - Index of one noraml/attack flow
    in test dataset """

    attack = ["Normal", "Reconnaissance"]
    train = train.reshape(len(train), len(columns))
    test = test.reshape(len(test), len(columns))

    # Create LIME explainer
    explainer = lime.lime_tabular.LimeTabularExplainer( train,
                                                        feature_names=
    columns,
                                                        class_names=
    attack,

    discretize_continuous=True)

    # Make prediction
    exp_normal = explainer.explain_instance(test[normal_indx] ,
                                            model.predict,
                                            num_features=len(columns),
                                            labels=(0,),
                                            top_labels=1)
    exp_attack = explainer.explain_instance(test[attack_indx] ,
                                            model.predict,
                                            num_features=len(columns),
                                            labels=(0,),
                                            top_labels=1)

    # Save prediction
    exp_normal.save_to_file(dnn_model_save_dir+"lime_prediction_normal.
    html")
    exp_attack.save_to_file(dnn_model_save_dir+"lime_prediction_attack.
    html")

    map = exp_attack.as_map()
    for i in map:
        print(i)

def create_dnn_model(ant_kolonner, activation='sigmoid'):
    model = Sequential()

    model.add(Dense(1024,input_dim=ant_kolonner,activation='relu'))
    model.add(Dropout(0.01))
    model.add(Dense(768,activation='relu'))
    model.add(Dropout(0.01))
    model.add(Dense(512,activation='relu'))
    model.add(Dropout(0.01))
    model.add(Dense(1))
    model.add(Activation(activation))
    model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])

    return model

def predict_dnn(model, testdata, testlabel, columns, combined_data):
    """ numb_pred : number of predictions"""

    xai_common(model, testdata, testlabel, columns.drop('Class'), combined_data)

    print(testdata.shape)
    numb_row = testdata.shape[0]
    predict_all = model.predict(testdata.reshape(numb_row, len(columns)-1))
    predict_all_binary = np.greater(predict_all, .5)

    # Create metrics
    score = metrics.accuracy_score(predict_all_binary, testlabel)
    print("Validation score: {}".format(score, '.3f'))

def one_hot(df, cols):
    """
    Source: https://www.kaggle.com/code/razor08/unsw-categorical-final/notebook
    Author: Jay Sinha, M. Manollas
    Date: 13. Dec, 2021

    One-hot encoding
    @param df pandas DataFrame
    @param cols a list of columns to encode
    @return a DataFrame with one-hot encoding
    """
    for each in cols:
        dummies = pd.get_dummies(df[each], prefix=each, drop_first=False)
        df = pd.concat([df, dummies], axis=1)
        df = df.drop(each, axis=1)
    return df

def normalize(df, cols):
    """
    Source: https://www.kaggle.com/code/razor08/unsw-categorical-final/notebook
    Author: Jay Sinha, M. Manollas
    Date: 13. Dec, 2021

    Function to min-max normalize
    @param df pandas DataFrame
    @param cols a list of columns to encode
    @return a DataFrame with normalized specified features
    """
    result = df.copy() # do not touch the original df

    for feature_name in cols:
        max_value = df[feature_name].max()
        min_value = df[feature_name].min()
        if max_value > min_value:
            result[feature_name] = (df[feature_name] - min_value) / (
    max_value - min_value)
    return result

def preprocess_unsw_remove_attacks(df, keep):
    remove_attacks = ['Analysis', 'Backdoor', 'DoS', 'Exploits', 'Fuzzers', 'Generic','Reconnaissance', 'Shellcode', 'Worms']
    # Keep this one
    remove_attacks.remove(keep)

    for i in remove_attacks:
        df.drop(df.index[df['attack_cat'] == i], inplace=True)

    return df

def preprocess_unsw_nb15_sigmoid(combined_data):
    tmp_label = combined_data.pop('label')

    cols = ['proto','state','service']

    combined_data_hot = one_hot(combined_data,cols)

    new_train_df = normalize(combined_data_hot,combined_data_hot.columns)
    new_train_df["Class"] = tmp_label

    # Brukt for aa sende un-normalized data til shap analyse
    combined_data_hot["Class"] = tmp_label

    y_train=new_train_df["Class"]
    combined_data_X = new_train_df.drop('Class', 1)

    oversample = RandomOverSampler(sampling_strategy='minority')

    kfold = StratifiedKFold(n_splits=2,shuffle=True,random_state=42)
    kfold.get_n_splits(combined_data_X,y_train)

    model = create_dnn_model(np.size(combined_data_X,1), activation='sigmoid')


    # Split dataen i to og tren modellen
    """
    Source: https://www.kaggle.com/code/razor08/unsw-categorical-final/notebook
    Author: Jay Sinha, M. Manollas
    Date: 13. Dec, 2021
    """
    for train_index, test_index in kfold.split(combined_data_X,y_train):
        train_X, test_X = combined_data_X.iloc[train_index], combined_data_X.iloc[test_index]
        train_y, test_y = y_train.iloc[train_index], y_train.iloc[test_index]

        train_X_over,train_y_over= oversample.fit_resample(train_X, train_y)

        x_columns_train = new_train_df.columns.drop('Class')
        x_train_array = train_X_over[x_columns_train].values
        x_train_1=np.reshape(x_train_array, (x_train_array.shape[0], x_train_array.shape[1], 1))

        y_train_1 = train_y_over

        x_columns_test = new_train_df.columns.drop('Class')
        x_test_array = test_X[x_columns_test].values
        x_test_2=np.reshape(x_test_array, (x_test_array.shape[0], x_test_array.shape[1], 1))

        y_test_2 = test_y

        if skip_training:
            # Load model
            model = keras.models.load_model(dnn_model_save_dir+"dnn_sigmoid_model.hdf5")
        else:
            model = train_dnn_model(model, x_train_1, y_train_1, x_test_2=x_test_2, y_test_2=y_test_2)

        # Run XAI and prediction anlaysis
        predict_dnn(model,x_test_2, y_test_2, new_train_df.columns, combined_data_hot)



def parsing():
    global data_unsw
    global skip_training
    global epochs
    global skip_remove_attacks

    parser = argparse.ArgumentParser(description="DNN utilising sigmoid (Binary)")
    parser.add_argument('-d','--dataset', help='Folder where the following datasets are stored: UNSW_NB15_testing-set.csv, UNSW_NB15_training-set.csv')
    parser.add_argument('-p','--predict', help='Set argument to only run prediction, load an already trained model',
                        action="store_true")
    parser.add_argument('-s','--skippre', help='Use a cleaned out dataset with only reconnaissance and normal traffic',
                        action="store_true")
    parser.add_argument('-e', '--epochs', help='Number of epochs')
    args = parser.parse_args()

    try:
        if args.dataset:
            data_unsw = args.dataset
            print("New dataset stored location:", data_unsw)
    except: pass
    try:
        if args.predict:
            skip_training = True
            print("Skipping training, only predicting")
    except: pass
    try:
        if args.skippre:
            skip_remove_attacks = True
            print("Skipping removing attacks")
    except: pass
    try:
        if args.epochs:
            epochs = int(args.epochs)
            print("Running {} epochs".format(epochs))
```

```python
    except: pass

def dataset_analysis(td):
    print(td.attack_cat.value_counts())
    # 1 og 0, bruk den for binaer klassifisering
    print(td.label.value_counts())
    print(td.columns)


def main():
    """ Parse input """
    parsing()

    if (skip_remove_attacks):
        combined_data = pd.read_csv(removed_attacks_csv, index_col=[0])
    else:
        """Les inn dataset"""
        td = pd.read_csv(data_unsw+nb_train)
        ts = pd.read_csv(data_unsw+nb_test)

        """Enkel analyse av dataset"""
        #dataset_analysis(td)

        """Preprosesser"""
        td = preprocess_unsw_remove_attacks(td, 'Reconnaissance')
        ts = preprocess_unsw_remove_attacks(ts, 'Reconnaissance')

        # Dont need id
        td = td.drop('id', axis=1)
        ts = ts.drop('id', axis=1)


        combined_data = pd.concat([td,ts])

    preprocess_unsw_nb15_sigmoid(combined_data)

if __name__ == "__main__":
    main()
```

B Interview - English translated version

B.1 Introduction



2022 Alarm Interview – Translated English

- Håkon's master thesis with the university of Oslo

Preparation

Central: - What do you need to make a good alarm assessment - How does signature vs xai measure against each other What is the goal? Get answers to problems - «What is considered as a good alarm for a SOC?» - - How to show good alarms with XAI	The starting point for the interview? - Theoretical foundation - Want to investigate whether what the experts think is important with an alarm - Want to ask the experts about what may be missing with current signature-based and ML-based alarms, which would help them in the assessments Given signature, what is ML missing Link ML + XAI - Analyst's assessment

Checklist before interview:
- Excel sheet so you can note if privacy is an issue
- Print some copies of it below you can look at during the interview
- Get water / coffee / snacks

Remember:
- The interview is a conversation, DO NOT interrogate or exam (nod and smile)
- Follow-up question: How? Can you give an example?

Introduction:

Presentation of me who will conduct the interview: Name, place of residence, education and job
What the interview is about: Master's thesis what is an explanatory alarm, how 2 methods within explanatory ML in anomaly-directed alarm generation will be able to come into play.
The interviewees will be anonymous, some introductory questions about education and work experience.
Can withdraw from the interview and study at any time, contact me if desired

Is it okay if I record the interview? It will only be used to facilitate the analysis work afterwards and will be deleted after the master's thesis is completed.

(Read and Sign Consent Form)
(START RECORDING)

"X, this is interview number X" (start recording with this, so it will be easy to sort those, where x is the interview number)

B.2 Introduction

Getting to know the analyst

Education	Higher education - bachelor, master, relevant bachelor - relevant master
Can you tell me briefly about your work tasks?	
Work experience in this field?	Little Medium Large
Experience with KI, ML?	Little Medium Large

Main part:

- How would you define an alarm?
- How much of your time is spent handling alarms?
- Do you handle network-based alarms?
- Do you handle client-based alarms?

- We now isolate the description of an alarm to the alert that appears, with time, name / type and the devices involved (In the form of IP addresses).

- If you are going to think about what a good alarm is. What do you think is important?
- In what way can an alarm be understood (That you understand what it is about, and can act)?

Objectives of alarm assessment:
- What will help you **Correlate** the alarm?
- What will help you to **classify** the alarm?

B.3 Case - Signature

Alarm evaluation:

- The team you are part of will set up a signature-based method, and a machine learning-based method on their network, to detect traffic of the reconnaissance type.

- The reconnaissance traffic you are going to detect is like the tool nmap produces (TCP / UDP scanning, service / OS detection).

- With the signature-based method, you have created a suricata signature that detects SYN scanning, by looking at the flag that has been set, and counting the frequency from a transmitter.

- With the ML-based, you have taken a clip of the normal traffic to the company, run nmap, and combined it into one data set, on which a machine learning model has been trained. This says if something is of the type "normal" or "attack".

- You will see the following alarms on the screen, which we will go through, before I will ask some questions related to them.

- 10.0.0.100 is a well-known network over which you have control.

(Go over signature, ML, additional note and normal picture)

SIGNATURE

Varsel raw:

```
03/09/2022-20:37:49.833584  [**] [1:2300000:3] Reconnaissance with nmap's SYN SCAN [**]
[Classification: Attempted Information Leak] [Priority: 10] {TCP} 10.0.0.99:60522 -> 10.0.0.100:1309
```

Alarm pretty:

Tidspunkt	SID	Navn	Klassifisering	Pri	Prot.	Fra -> Til
03/09/2022-20:37:49.8	2300000	Reconnaissance with nmap's SYN SCAN	Attempted Information Leak	10	TCP	10.0.0.99:60522 -> 10.0.0.100:1309

Regelen (viktige flow-verdier):

```
alert tcp any any -> any any (msg:" Reconnaissance with nmap's SYN SCAN"
flow:stateless; flags:S,12; classtype:attempted-recon;sid:2300000; priority:10; rev:1; threshold:type
threshold, track by_src, count 50, seconds1;)
```

Signature explanation:

- Alert
- Tcp
- Any any -> any any
- Msg
- Flow:stateless
- Flags:S,12
- Classtype:attempted-recon
- Sid:2300000
- Priority
- Rev:1
- Threshold:type threshold, track by_src, count 50, seconds1

B.4 Case - Anomaly: overview

MACHINE LEARNING

Alarm raw:

03/09/2022-20:37:49.833584 [**] [ML_ANN_R] Possible reconnaissance activity [**] [Classification: Attempted Information Leak] [Priority: 10] {TCP} [3f9a:9e65::e118:4f87]:60522 -> 10.0.0.100:1309

Alarm pretty:

Tidspunkt	SID	Navn	Klassifisering	Pri	Prot	Fra -> Til
03/09/2022-20:37:49.8	ML_ANN_R	Possible reconnaissance activity	Attempted Information Leak	10	TCP	[3f9a:9e65::e118:4f87]: 60522 -> 10.0.0.100:1309

Rule description (example of textual description):

Legitimate = Not the reconnaissance we are looking for

The alarm is triggered because:

- The protocol is ipv6-no.
  - 0% of all legitimate traffic has this value, while 0.05% of all reconnaissance uses this value.
- TTL from source to destination is 254.
  - 28% of all legitimate traffic has this value, while 99.5% of all reconnaissance use this value.
- Source's TCP windows advertisement Value is 0.
  - 28% of all legitimate traffic has this value, while 50% of all reconnaissance use that value.

Flow data

To provide an overview of the flow information, see the additional note. It provides an overview of all the flow columns with a brief description.

XAI methods

Below, two different algorithms called SHAP and LIME are used, which try to say something about which flow values had the most influence on the model classifying it as "reconnaissance".
Here, only the top 3 columns / properties that received the highest score (importance) were selected.

SHAP

Flow column	Flow value	Important (SUM 8.2)	Description
proto_ipv6-no	Ipv6-no	0.18	No next header for IPv6
sttl	254	0.14	Source to destination TTL
swin	0	0.09	Source TCP window advertisement value

LIME

Flow column	ML value	Flow value	Important	Description
proto_ipv6-no > 0	1	Ipv6-no	0.18	No next header for IPv6
0.12 < sttl <=1	1	254	0.12	Source to destination TTL
Swin <= 0	0	0	0.07	Source TCP window advertisement value

B.5 Case - Anomaly: regular traffic

Description of the columns:

- **Flow column**: The name of the column (LIME: which value must be in «ML value» for it to consider the data point as reconnaissance)
- **ML Value**: The value ML gets AFTER preprocessing / washing of the data
- **Flow value**: Original how the data looks BEFORE preprocessing / washing
- **Importance**: The value SHAP / LIME has given
- **Assessment**: Description of the flow column

Understanding regular traffic:

- Here is an attempt to better understand the normal picture of the company, and how attacks appear in the data. The starting point here is the dataset the team took out to train the anomaly model.
- For each flow column mentioned above, the number of attacks accumulates against normal ones that have unique values in those columns.
- Ex. "Proto ipv6-no" has 2 unique values (0 and 1). Then the number of attacks and normal are displayed with these respective values.

Total normal: 93007
Total Attack: 13980

Column	Value	Normal	Attack
Proto_ipv6-no			
	0	93007	13973
	1	0	7
Sttl			
	255	2	1
	254	26279	13922
	252	2	0
	64	181	0
	63	32	0
	62	6296	40
	60…1*	56352	0
	0	3846	24
Swin			
	255	66949	6965
	245…5*	20	0
	0	26031	7022

* Represents an unspecified collection of unique values from and including given values on each side of ‘…’. These are collected since there are no attacks within that range of values.

B.6 Case - Anomaly: Feature overview

Additional note 1: The data collected and used for training in the model

The data can be divided into 5 groups
1. Flow data: Has the identification-focused attributes between devices
2. Basic data: Contains the attributes that represent the connection protocol
3. Content data: Contains attributes for TCP / IP and some attributes for http services
4. Time data: Attributes about time, such as the time interval between packets, start and end time, RTT for TCP
5. Additional data: Attributes to protect the service of the protocols, and attributes created by looking at the previous 100 connections

#	Navn	Beskrivelse
	FLOW DATA	
1	Srcip	Source IP address
2	Sport	Source port number
3	Dstip	Destinations IP address
4	Dsport	Destination port number
5	Proto	Protocol type (TCP, UDP…)
	BASIC DATA	
6	State	The states and its dependent protocol e.g., CON.
7	Dur	Row total duration.
8	sbytes	Source to destination bytes.
9	dbytes	Destination to source bytes.
10	Sttl	Source to destination time to live.
11	dttl	Destination to source time to live.
12	sloss	Source packets retransmitted or dropped.
13	dloss	Destination packets retransmitted or dropped.
14	service	Such as http, ftp, smtp, ssh, dns and ftp data.
15	sload	Source bits per second.
16	dload	Destination bits per second.
17	spkts	Source to destination packet count.
18	dpkts	Destination to source packet count.
	INNHOLD DATA	
19	swin	Source TCP window advertisement value.
20	dwin	Destination TCP window advertisement value.
21	Stcpb	Source TCP base sequence number.
22	dtcpb	Destination TCP base sequence number.
23	smeansz	Mean of the packet size transmitted by the srcip.
24	dmeansz	Mean of the packet size transmitted by the dstip.
25	trans_depth	The connection of http request/response transaction.
26	res_bdy_len	The content size of the data transferred from http.
	TIDS DATA	
27	sjit	Source jitter.
28	djit	Destination jitter.
29	stime	Row start time.
30	ltime	Row last time.
31	sintpkt	Source inter-packet arrival time packet arrival time.
32	dintpkt	Destination inter packet arrival time.
33	tcprtt	Setup round trip time, the sum of ’synack’ and ’ackdat’.
34	synack	The time between the SYN and the SYN_ACK packets.
35	ackdat	The time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	If srcip (1) = dstip (3) and sport (2) = dsport (4), assign 1 else 0.

B.7 Case - Anomaly: Feature overview

	EKSTRA DATA	
37	ct_state_ttl	No. of each state (6) according to values of sttl (10) and dttl (11).
38	ct_flw_http_mthd	No. of methods such as Get and Post in http service.
39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	No of flows that has a command in ftp session.
41	ct_srv_src	No. of rows of the same service (14) and srcip (1) in 100 rows.
42	ct_srv_dst	No. of rows of the same service (14) and dstip (3) in 100 rows.
43	ct_dst_ltm	No. of rows of the same dstip (3) in 100 rows.
44	ct_src_ ltm	No. of rows of the srcip (1) in 100 rows.
45	ct_src_dport_ltm	No of rows of the same srcip (1) and the dsport (4) in 100 rows.
46	ct_dst_sport_ltm	No of rows of the same dstip (3) and the sport (2) in 100 rows.
47	ct_dst_src_ltm	No of rows of the same srcip (1) and the dstip (3) in 100 records.

Additional information on Ipv6-no:

- IPv6 can have many "extension headers", which are placed between the standard / static header, and the header for protocols on higher layers.
- IPv6's static header has a field called «Next header», where you can specify which type of extensions you use, and which thus follows directly after the static header.

Value 59 (No next header) in the ipv6 protocol's next header field »is set.

- It indicates that there are no headers following this.
- Not even a header for protocols on the teams above.
- This means that from the header's point of view, the IPv6 package ends right after it, ie no payload.
- There may be data in the payload if the length of the first header of the package is greater than the length of all additional headers.
- That data should then be ignored by the host but forwarded unchanged by routers.

Additional information on swin:

- Also called TCP receiver window size, is an information about how much data (in bytes) the receiver is willing to receive. The receiver can use this value to control how much data it receives.
- In our case, it is the source that sets it, and it is set to 0
- 0 from a client, will cause data transfer to be paused from the server side, until the one from the client receives a TCP window update package.

I am not an expert on all the details in each of these columns, but if something feels interesting beyond what is described, we google it.

B.8 Case - Questions

Questions for alarm assessment:

- How do you experience that these methods differ from each other if you were to assess the alarm?
- Is there something you experience that both are missing, so that you can make a good assessment?
- Do you have an idea of how you could show the ML assessment in a different way, so that it would have been better?
  - o What about the section under «rule description»?
- Can you imagine a workflow where the methods from LIME and SHAP are used?
  - o As part of getting to know new attacks, do you think the methods would contribute insight to an analyst?
  - o Do you think the methods can support signature writing?

General about Anomaly

- A known challenge for analysts who are not used to anomaly-based detection is to relate to the fact that not all anomalies are "malicious", but all signature alarms are associated with harmful behavior.
  - o Do you have any thoughts on how to adapt the analyst more easily to this issue?
- One possible feature in anomaly detection and machine learning based is an "alarms per IP", which helps the analyst to keep an eye on interesting, but not necessarily malicious IPs over time to confirm or disprove a suspicion.
  - o What are your thoughts on this feature?

Assessment of statements

If you were to give each of the statements an assessment from little-medium-much, which would you have given:

	Signature	Anomaly
It helps to understand the alarm		
The information shortens the assessment time		
The information requires a lot from the analyst		

Finally:

- Summary of what has been said
- Clarify ambiguities

- Something extra you think might be relevant, or something you want to add?

- In conclusion, if you were to give two pieces of advice to someone who is developing anomaly solutions for an SOC, what would it be?
- Thank you for participating 😊

C Interview - Norwegian original version

2022 Alarm Intervjue – Original Norwegian

- Håkons masteroppgave ved universitetet i Oslo

Forarbeid

Sentrale: - Hva trenger man for å gjøre en god alarmvurdering - Hvordan måler signatur vs xai mot hverandre Hva er målet? Få svar på problemstilling - «What is considered as a good alarm for a SOC?» - How to show good alarms with XAI	Utgangspunktet for intervjuet? - Teorifundament - Ønsker å undersøke om hva ekspertene synes er viktig med en alarm - Ønsker å spørre ekspertene om hva som kan mangle med nåværende signaturbaserte og ML baserte alarmer, som ville hjulpet de i vurderingene 1. Gitt signatur, hva mangler ML 2. Mappe ML + XAI – Analytiker sin vurdering

Sjekkliste før intervjue:

- Excel ark så du kan notere om personvern blir en utfordring
- Skriv ut noen kopier av det under dere kan se på under intervjuet
- Hent vann/kaffe/snacks

Husk:

- Intervjuet er en samtale, IKKE avhør elle eksamen (nikk og smil)
- Oppfølgingsspørsmål: Hvordan? Kan du gi et eksempel?

Introduksjon:

Presentasjon av meg som skal gjennomføre intervjuet: Navn, bosted, utdanning og jobb
Hva intervjuet handler om: Masteroppgave hva som er en forklarbar alarm, hvordan 2 metoder innen forklarbar ML i anomalirettet alarmgenerering vil kunne spille inn.
Intervjuobjektene vil være anonyme, noen innledende spørsmål om utdanning og arbeidserfaring.
Kan trekke deg fra intervjuet og studien når som helst, kontakt meg hvis ønskelig

Går det fint om jeg tar opptak av intervjuet? Det skal bare brukes for å lette analysearbeidet i ettertid, og vil slettes etter masteroppgaven er ferdig.

(Les, og Signer samtykkeskjema)
(START OPPTAK)

«X, dette er intervjue nummer X» (start opptaket med dette, så det blir enkelt å sortere de, der x er intervjuenummeret)

Bli kjent med analytikeren

Utdanning	Videregående - bachelor, master, relevant bachelor - relevant master
Kan du fortelle meg kort om dine arbeidsoppgaver?	
Arbeidserfaring innen dette feltet?	Lite Middels Mye
Erfaring med KI, ML?	Lite Middels Mye

Hoveddel:

- Hvordan vil du definere en alarm?
- Hvor mye av tiden din går til å håndtere alarmer?
- Håndterer du nettverks-baserte alarmer?
- Håndterer du klient-baserte alarmer?

- Vi isolerer nå beskrivelsen av en alarm til varselet som dukker opp, med tidspunkt, navn/type og de involverte enhetene (I form av IP adresser).

- Hvis du skal tenke på hva som er en god alarm. Hva tror du er viktig?
- På hvilken måte kan en alarm gjøre seg forstått (At du skjønner hva den går ut på, og kan agere)?

Mål med alarmvurdering:
- Hva vil hjelpe deg med å **Korrelere** alarmen?
- Hva vil hjelpe deg til å **klassifisere** alarmen?

Alarmvurdering:

- Teamet du er en del av skal sette opp en signaturbasert metode, og en maskinlæringbasert metode på nettverket deres, for å detektere trafikk av typen rekognisering.
- Rekogniseringstrafikken dere skal detektere ligner på den verktøyet nmap produserer (TCP/UDP scanning, service/OS deteksjon).

- Med den signaturbaserte metoden har dere laget en suricata-signatur som detekterer SYN scanning, ved å se på flagget som er satt, og å telle hyppighet fra en sender.
- Med den MLbaserte har dere tatt et utklipp av normaltrafikken til bedriften, kjørt nmap, og kombinert det til ett datasett, som en maskinlæringmodell er blitt trent på. Denne sier om noe er av typen «normal» eller «angrep».

- Du får opp følgende varsler på skjermen, som vi skal gå igjennom, før jeg vil stille noen spørsmål tilknyttet dem.

- 10.0.0.100 er et kjent nettverk dere har kontroll over.

(Gå over signatur, ML, tilleggsnotat og normalbildet)

SIGNATUR

Varsel raw:

```
03/09/2022-20:37:49.833584  [**] [1:2300000:3] Reconnaissance with nmap's SYN SCAN [**]
[Classification: Attempted Information Leak] [Priority: 10] {TCP} 10.0.0.99:60522 -> 10.0.0.100:1309
```

Varsel pretty:

Tidspunkt	SID	Navn	Klassifisering	Pri	Prot.	Fra -> Til
03/09/2022-20:37:49.8	2300000	Reconnaissance with nmap's SYN SCAN	Attempted Information Leak	10	TCP	10.0.0.99:60522 -> 10.0.0.100:1309

Regelen (viktige flow-verdier):

```
alert tcp any any -> any any (msg:" Reconnaissance with nmap's SYN SCAN"
flow:stateless; flags:S,12; classtype:attempted-recon;sid:2300000; priority:10; rev:1; threshold:type
threshold, track by_src, count 50, seconds1;)
```

Regelforklaring:

- Alert
- Tcp
- Any any -> any any
- Msg
- Flow:stateless
- Flags:S,12
- Classtype:attempted-recon
- Sid:2300000
- Priority
- Rev:1
- Threshold:type threshold, track by_src, count 50, seconds1

MASKINLÆRING

Varsel raw:

03/09/2022-20:37:49.833584 [**] [ML_ANN_R] Possible reconnaissance activity [**] [Classification: Attempted Information Leak] [Priority: 10] {TCP} [3f9a:9e65::e118:4f87]:60522 -> 10.0.0.100:1309

Varsel pretty:

Tidspunkt	SID	Navn	Klassifisering	Pri	Prot	Fra -> Til
03/09/2022-20:37:49.8	ML_ ANN_R	Possible reconnaissance activity	Attempted Information Leak	10	TCP	[3f9a:9e65::e118:4f87]: 60522 -> 10.0.0.100:1309

Regel-beskrivelse (eksempel på tekstlig beskrivelse):

Legitim = Ikke rekogniseringen vi ser etter

Alarmen trigger på grunn av at:

- Protokollen er ipv6-no.
  - 0% av all legitim trafikk har denne verdien, mens 0,05% av all rekognisering benytter den verdien.
- TTL fra source til destination, er på 254.
  - 28% av all legitim trafikk har denne verdien, mens 99,5% av all rekognisering benytter den verdien.
- Source sin TCP windows advertisement Verdi er på 0.
  - 28% av all legitim trafikk har denne verdien, mens 50% av all rekognisering benytter den verdien.

Flow-dataen

For å gi en oversikt på flow-informasjonen, se tilleggsnotatet. Den gir en oversikt over alle flow kolonnene med en kort beskrivelse.

XAI metoder

Under er det benyttet to forskjellige algoritmer kalt SHAP og LIME, som prøver å si noe om hvilke flow-verdier som hadde mest innflytelse på at modellen klassifiserte det som «rekognisering». Her er bare de top 3 kolonnene/egenskapene som fikk størst score (viktighet) valgt frem.

SHAP

Flow kolonne	Flow verdi	Viktighet (SUM 8.2)	Beskrivelse
proto_ipv6-no	Ipv6-no	0.18	No next header for IPv6
sttl	254	0.14	Source to destination TTL
swin	0	0.09	Source TCP window advertisement value

LIME

Flow kolonne	ML verdi	Flow verdi	Viktighet	Beskrivelse
proto_ipv6-no > 0	1	Ipv6-no	0.18	No next header for IPv6
0.12 < sttl <=1	1	254	0.12	Source to destination TTL
Swin <= 0	0	0	0.07	Source TCP window advertisement value

Beskrivelse av kolonnene:

- **Flow kolonne:** Navnet på kolonnen (LIME: hvilke verdi som skal stå i «ML verdi» for at den skal vurdere datapunktet som rekognisering)
- **ML Verdi:** Verdien ML får ETTER preprosessering/vasking av dataen
- **Flow verdi:** Originalt hvordan dataen ser ut FØR preprosessering/vasking
- **Viktighet:** Verdien SHAP/LIME har gitt
- **Vurdering:** Beskrivelse av flow kolonnen

Forståelse av normalbildet:

- Her er et forsøk på å forstå normalbildet til bedriften bedre, og hvordan angrep viser seg i dataen. Utgangspunktet her er datasettet teamet hentet ut for å trene anomali modellen.
- For hver flow kolonne nevnt over, er antall angrep akkumuler mot normale som har unike verdier i de kolonnene.
- Eks. «Proto ipv6-no» har 2 unike verdier (0 og 1). Så vises antallet angrep og normal med disse respektive verdiene.

Totalt normal: 93007
Totalt angrep: 13980

Kolonne	**Verdi**	**Normal**	**Angrep**
Proto_ipv6-no			
	0	93007	13973
	1	0	7
Sttl			
	255	2	1
	254	26279	13922
	252	2	0
	64	181	0
	63	32	0
	62	6296	40
	60…1*	56352	0
	0	3846	24
Swin			
	255	66949	6965
	245…5*	20	0
	0	26031	7022

* Representerer en uspesifisert samling av unike verdier fra og med, til og med gitte verdier på hver side av '…'. Disse er samlet siden det ikke er noen angrep innen den rekken av verdier.

Tilleggsnotat 1: Dataen som er samlet inn og brukt for trening i modellen

Dataen kan deles inn i 5 grupper

1. Flow data: Har de identifiseringsfokuserte attributtene mellom enheter
2. Basic data: Inneholder attributtene som representerer tilkoblingsprotokollen
3. Innhold data: Inneholder attributter for TCP/IP og noen attributter for http tjenester
4. Tids data: Attributter om tid, som tidsintervallet mellom pakker, start og slutt tid, RTT for TCP
5. Ekstra data: Attributter som skal beskytte tjenesten til protokollene, og attributter laget ved å se på de 100 forrige tilkoblingene

#	Navn	Beskrivelse
	FLOW DATA	
1	Srcip	Source IP address
2	Sport	Source port number
3	Dstip	Destinations IP address
4	Dsport	Destination port number
5	Proto	Protocol type (TCP, UDP…)
	BASIC DATA	
6	State	The states and its dependent protocol e.g., CON.
7	Dur	Row total duration.
8	sbytes	Source to destination bytes.
9	dbytes	Destination to source bytes.
10	Sttl	Source to destination time to live.
11	dttl	Destination to source time to live.
12	sloss	Source packets retransmitted or dropped.
13	dloss	Destination packets retransmitted or dropped.
14	service	Such as http, ftp, smtp, ssh, dns and ftp data.
15	sload	Source bits per second.
16	dload	Destination bits per second.
17	spkts	Source to destination packet count.
18	dpkts	Destination to source packet count.
	INNHOLD DATA	
19	swin	Source TCP window advertisement value.
20	dwin	Destination TCP window advertisement value.
21	Stcpb	Source TCP base sequence number.
22	dtcpb	Destination TCP base sequence number.
23	smeansz	Mean of the packet size transmitted by the srcip.
24	dmeansz	Mean of the packet size transmitted by the dstip.
25	trans_depth	The connection of http request/response transaction.
26	res_bdy_len	The content size of the data transferred from http.
	TIDS DATA	
27	sjit	Source jitter.
28	djit	Destination jitter.
29	stime	Row start time.
30	ltime	Row last time.
31	sintpkt	Source inter-packet arrival time packet arrival time.
32	dintpkt	Destination inter packet arrival time.
33	tcprtt	Setup round trip time, the sum of 'synack' and 'ackdat'.
34	synack	The time between the SYN and the SYN_ACK packets.
35	ackdat	The time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	If srcip (1) = dstip (3) and sport (2) = dsport (4), assign 1 else 0.

	EKSTRA DATA	
37	ct_state_ttl	No. of each state (6) according to values of sttl (10) and dttl (11).
38	ct_flw_http_mthd	No. of methods such as Get and Post in http service.
39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	No of flows that has a command in ftp session.
41	ct_srv_src	No. of rows of the same service (14) and srcip (1) in 100 rows.
42	ct_srv_dst	No. of rows of the same service (14) and dstip (3) in 100 rows.
43	ct_dst_ltm	No. of rows of the same dstip (3) in 100 rows.
44	ct_src_ ltm	No. of rows of the srcip (1) in 100 rows.
45	ct_src_dport_ltm	No of rows of the same srcip (1) and the dsport (4) in 100 rows.
46	ct_dst_sport_ltm	No of rows of the same dstip (3) and the sport (2) in 100 rows.
47	ct_dst_src_ltm	No of rows of the same srcip (1) and the dstip (3) in 100 records.

Ekstra informasjon på **Ipv6-no**:

- IPv6 kan ha mange «extension headere», som legges mellom standard/statisk header, og headeren for protokoller på høyere lag.
- IPv6 sin statiske header har et felt kalt «Next header», hvor man kan spesifisere hvilke type extensions man benytter, og som dermed følger rett etter den statiske headeren.
- Verdi 59 (No next header) i ipv6 protokollen sin «next header field» er satt.
- Den indikerer at det ikke er noen headere som etterfølger denne.
- Selv ikke en header for protokoller på lagene over.
- Det betyr at fra headerens synspunkt så slutter IPv6 pakken rett etter den, altså ingen payload.
- Det kan ligge data i payloaden om lengden i den første headeren av pakken er større enn lengden av alle tilleggsheaderene.
- Den dataen skal da ignoreres av hosten, men videresendes uendret av rutere.

Ekstra informasjon på **swin**:

- Også kalt TCP receiver window size, er en informering om hvor mye data (i bytes) mottakeren er villig til å motta. Mottakeren kan benytte denne verdien til å kontrollere hvor mye data den får.
- I vårt tilfelle er det source som setter den, og den er satt til 0
- 0 fra en klient, vil gjøre at data-overføring settes på pause fra serversiden, helt til den fra klienten mottar en TCP window update pakke.

Jeg er ingen ekspert på alle detaljene i hver og en av disse kolonnene, men om noe oppleves interessant utover det som er beskrevet, så googler vi det.

Spørsmål til alarmvurdering:

- Hvordan opplever du at disse metodene skiller seg fra hverandre om du skulle gjort en vurdering av alarmen?
- Er det noe du opplever at begge mangler, for at du kan gjøre en god vurdering?
- Har du en formening om hvordan man kunne vist ML vurderingen på en annen måte, for at det hadde blitt bedre?
  - Hva med avsnittet under «regel-beskrivelse»?
- Kan du ser for deg en arbeidsflyt hvor metodene fra LIME og SHAP blir benyttet?
  - Som en del av å bli kjent med nye angrep, tror du metodene ville bidratt med innsikt til en analytiker?
  - Tror du metodene kan støtte signaturskriving?

Generelt om Anomali

- En kjent utfordring for analytikere som ikke er vant med anomalibasert deteksjon, er det å forholde seg til at ikke alle anomalier er «malicious», men alle signatur-alarmer er knyttet til skadelig oppførsel.
  - Har du noen tanker om hvordan man lettere kan tilpasse analytikeren for denne problemstillingen?
- En mulig funksjon innen anomalideteksjon og maskinlæringbasert er en «alarmer per IP», som bidrar til at analytikeren kan holde et øye med interessante, men ikke nødvendigvis malicious IPer over tid for å bekrefte eller avkrefte en mistanke.
  - Hva er dine tanker om denne funksjonen?

Vurdering av utsagn

Om du skulle gi hver av utsagnene en vurdering fra lite-middels-mye, hvilke hadde du gitt:

	Signatur	Anomali
Det bidrar med å forstå alarmen		
Informasjonen korter ned vurderingstiden		
Informasjonen krever mye av analytikeren		

Avslutningsvis:

- Oppsummering av det som er sagt
- Oppklare uklarheter

- Noe ekstra du tror kan være relevant, eller noe du ønsker å legge til?

- Avslutningsvis, hvis du skulle gi to råd til en som utvikler anomaliløsninger for en SOC, hva hadde det vært?
- Takke for deltagelsen 😊

D Consent form

Vil du delta i forskningsprosjektet

Masteroppgave for bruk av XAI til nettverskhåndtering

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å identifisere hva som er en god alarm. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål
Dette er en del av en masteroppgave ved universitetet i oslo, institutt for informatikk.

Hvem er ansvarlig for forskningsprosjektet?
Universitetet i Oslo er ansvarlig for prosjektet.
Håkon Svee Eriksson vil gjennomføre all forskning og dokumentering

Hvorfor får du spørsmål om å delta?
Du er valgt ut siden du kjenner Håkon Svee Eriksson, og har kunnskap om håndtering av nettverksalarmer.

Hva innebærer det for deg å delta?
Metoden du blir utsatt for er et intervjue, og hvis det går greit for deg, et opptak som slettes etter transkribering.

Det er frivillig å delta
Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger
Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Den eneste som vil ha tilgang til opptaket er student Håkon Svee Eriksson.
Tiltaket for å sikre at ingen andre får tilgang til opptaket er å ha det lagret på en enhet ikke tilkoblet et nettverk, samtidig som at under transkriberingen vil Håkon sitte isolert.

Deltagerne vil ikke kunne bli gjenkjent i publikasjonen.

Dine rettigheter
Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitetet i Oslo har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?
Hvis du har spørsmål til studien, eller ønsker å benytte deg av dine rettigheter, ta kontakt med:

- Universitetet i Oslo ved Audun Jøsang (josang@ifi.uio.no).
- Vårt personvernombud: Roger Markgraf-Bye (personvernombud@uio.no)

Hvis du har spørsmål knyttet til NSD sin vurdering av prosjektet, kan du ta kontakt med:

- NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Med vennlig hilsen

Audun Jøsang	Håkon Svee Eriksson
(Forsker/veileder)	

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet *Masteroppgave for bruk av XAI til nettverkshåndtering*, og har fått anledning til å stille spørsmål. Jeg samtykker til:

□ å delta i intervjue

□ at mine personopplysninger lagres etter transkribering er ferdig, ikke senere enn levering av masteroppgaven 23 mai 2022

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

----

(Signert av prosjektdeltaker, dato)