# Towards XAI in the SOC – a user centric study of explainable alerts with SHAP and LIME

1st Håkon Svee Eriksson
*University of Oslo*
Oslo, Norway
haakose@uio.no

2nd Gudmund Grov
*Norwegian Defence Research Establishment & University of Oslo*
Kjeller, Norway
Gudmund.Grov@ffi.no

***Abstract*—Many studies of the adoption of machine learning (ML) in Security Operation Centres (SOCs) have pointed to a lack of transparency and explanation – and thus trust – as a barrier to ML adoption, and have suggested eXplainable Artificial Intelligence (XAI) as a possible solution. However, there is a lack of studies addressing to which degree XAI indeed helps SOC analysts. Focusing on two XAI-techniques, SHAP and LIME, we have interviewed several SOC analysts to understand how XAI can be used and adapted to explain ML-generated alerts. The results show that XAI can provide valuable insights for the analyst by highlighting features and information deemed important for a given alert. As far as we are aware, we are the first to conduct such a user study of XAI usage in a SOC and this short paper provides our initial findings.**

***Index Terms*—Interpretability, explainability, artificial intelligence, machine learning, security operation center, intrusion detection system, explainable artificial intelligence, user studies**

## I. Introduction

At the core of a Security Operation Centre (SOC) is analysts monitoring and analysing alerts. After an alert is deemed suspicious, a closer inspection is usually conducted by an experienced analyst, before closing or escalating the alert to an incident management team [14], [36]. Traditionally, signature-based intrusion detection systems (IDS), such as Suricata[1], have mostly been used to generate alerts from real-time monitoring of systems and networks. Such signatures are typically developed manually from cyber threat intelligence, published vulnerabilities (e.g. CVEs[2]) or analysis of known malware or incidents. When analysing an alert, the *source* of the signature can then provide necessary contextual information (or explanation) for the analyst. Examples of such 'source' can for instance be the threat intelligence report, or knowledge of the malware/incident, from which it was generated.

The resources required to develop and maintain signatures does not scale with the increased number of threats and its complexities. This has resulted in a stronger focus on data-driven detection techniques, where machine learning (ML) is often used. However, predictions made by modern ML-systems are generally known to be hard to explain, which is also the case for ML-generated alerts. To illustrate, instead of providing the analyst with the intuition behind an alert, e.g. in terms of a threat intelligence report, the analyst is given a number that indicates how far the given data is from what a ML-trained model perceives as normal. Such a lack of transparency and explanation of both the ML-models, and predictions they make, have led to the creation of a sub-field within artificial intelligence (AI) called eXplainable AI (XAI).

Several studies of SOC environments, including the use of AI in the SOC, have pointed towards adapting and using XAI techniques to support analysts [25], [7], [21], [9], [32], [2], [16], [26]. There have also been several attempts applying XAI to explain ML-generated alerts [39], [38], [28], [15], [20], [27], [34]. However, studies on how and if XAI actually provided the necessary explanation for a security analyst seems to be missing [11].

A tongue-in-cheek definition of XAI is "*a translation from one mathematical notation nobody understands to another mathematical notation nobody understands*". Although this "definition" is not to be taken seriously, it still has a valid point: XAI-methods are mainly developed to support data scientist in interpreting and debugging their ML-models. A seminal paper by Sommer & Paxson [33] from more than a decade ago, introduced what they called the *semantic gap* between the language used by data scientist and the language used by security analyst; this is still relevant today [32].

Today, ML-based IDS' are almost exclusively compared and contrasted based on their performance [1], [10], [12]. As most alerts are managed manually, and future security automation is unlikely to be end-to-end ML, we argue that usability aspects should also be taken into account; there may be cases where a ML-model that produces more false alerts is preferable over a better performing model, if the generated alerts are easier to understand and analyse (and possible to automate). A first step in achieving this vision, is to improve our understanding of what security analysts need from alerts, what constitutes a "good" alert, and to which degree existing XAI-methods meet such needs. To reflect this contextual need of end-users, we use the definition of XAI by Arrieta et al. [4]:

> *Definition 1 (eXplainable AI (XAI) [4]):* Given a certain audience, explainability refers to the details and reasons a model gives to make its functioning clear or easy to understand.

[1] https://suricata.io/.

[2] https://www.cve.org/.

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

DOI til publisert versjon/DOI to published version: 10.1109/BigData55660.2022.10020248

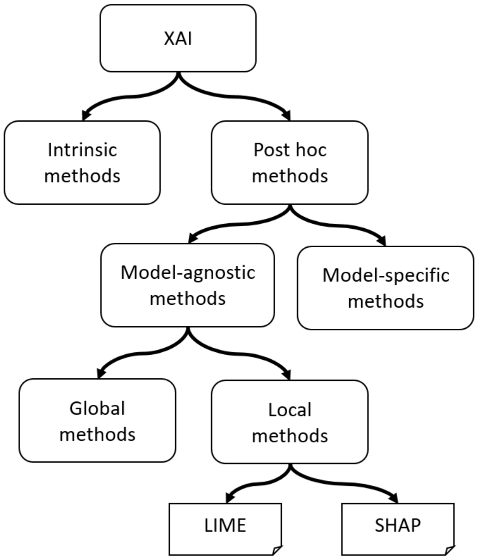

Fig. 1. XAI taxonomy (based on [22]).

Here, the context refers to a security analyst in a SOC, and explainability refers to how usable they are for the analyst.

Our long term vision is to build tools that can generate actionable alerts from ML-based models that meets the needs of security analysts. In this paper, we initiate the process of bridging XAI-based explanation of ML-generated IDS alerts with the needs of security analyst in a SOC, by reporting on the results from an initial five month experiment. We focus on the usage of two commonly used XAI methods, SHAP and LIME, used with a deep neural network on a well-known dataset, to generate explainable alerts from network traffic. As a baseline, a comparable Suricata signature was developed and used to generate a signature-based alert. These were used in interviews with SOC analysts to improve our understanding of their needs, using LIME and SHAP as a concrete case. The paper is based on the Master thesis by the first author [8], and is structured as follows:

- §II provides background on both XAI and existing user studies of SOCs;
- §III describes the experimental setup for the study, including alert generation and how the interview was conducted;
- §IV summarises the results from the interview;
- §V concludes and outlines further work.

## II. Background

We build on work that can be categorised along two lines of research: (A) XAI-methods with a focus on the use of ML-generated alerts, and (B) previous user studies of analysts in a SOC environment. [3]

### A. XAI

Figure 1 provides a taxonomy of XAI-approaches based on [22]. Here, we separate between ML-methods that are intrinsically explainable (e.g. logistic regression and decision trees) and methods that are explained post-hoc. Among post-hoc methods, we further separate model-specific methods, which only works for specific architectures and methods that are agnostic with respect to the underlying model. The focus of this paper is on model-agnostic methods, that we further split into global methods (explaining the underlying model), and local methods (explaining given predictions). Both SHAP and LIME falls under the latter category, and are one of the more common XAI-methods. The intuition behind our focus on local methods is that they are used to explain predictions (used to generate alerts), while we see global methods as support to data scientists for developing and debugging the ML-models.

| **Pred** | **S. IP** | **D. 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 | ... |
| ... | ... | ... | ... | ... | ... |

Fig. 2. Example of SHAP-generated datapoints.

*1) Local Interpretable Model-agnostic Explanations (LIME):* Given a model, LIME [19], [29] runs it multiple times for the same prediction, altering the feature inputs to see how each feature affects the output for each run. A new dataset with datapoints close to the datapoint in question is created, which is used as input. From these new datapoints, an interpretable model is created, weighted based on how close each datapoint in the new dataset is to the instance we are trying to explain.

*2) SHapley Additive exPlanations (SHAP):* SHAP [17] is based on Shapley values [31], originally developed for game theory, which shows how to distribute a *payout* evenly among the features based on their contribution to a prediction. The value is generated from the average of all marginal contributions of every coalition. The marginal contribution is the difference between two predictions, where one has changed a feature with a random valid value. Figure 2 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* (S. IP), the prediction is low, indicating that it is influential. The goal of SHAP is to interpret a prediction by calculating how each feature contributes to the overall score. The unique functionality of SHAP, contrary to Shapley values, is how the values are represented in an additive feature attribution method (linear model).

*3) Use of XAI in IDS:* There are several examples where XAI-methods have been used with ML-based IDS. Wawrowski et al. [39] used SHAP with a ML techniques called gradient boosting to implement an anomaly detection system. Mane & Rao [18] used a neural network to detect network intrusions, while presenting a XAI framework which explains each step of the ML pipeline. Global (post-hoc) explanations are provided to support the developer of the ML-model, while

[3] For a broader study of this topic, we refer to [8].

local explanations are provided both in terms of examples from the training set and post-hoc feature contributions (as with LIME and SHAP). Wang et al. [38] used SHAP's local capabilities to interpret single attack predictions, and global functionality to highlight important features. This combination tied feature values with various attack types. An approach first discussed in [38] used global post-hoc explanations as a form of enrichment, possibly gaining a deeper knowledge of attacks and their patterns. Finally, Mathews [20] used LIME to support explanation of classification of Windows malware from a deep neural network.

### B. User-centric SOC studies

There have been several studies focusing on the analyst in a SOC. Yu [25] targets human interaction aspects to elicit functional requirements for automating tasks performed by a digital *teammate* Feng et al. [9] developed a ML framework by generating labels from SOC notes in order to correlate IP-addresses, hosts and end-users. Akinrolabu et al. [2] discovered valuable features (for ML models) by interviewing SOC analysts. Oesch et al. [26] discovered several usability issues when using ML in a SOC. They also found that the analysts lacked an understanding of how scores were generated, resulting in misuse and mistrust. Alahmadi et al. [5] defined five properties to improve and speed up alert validation: Reliable, Explainable, Analytical, Contextual, and Transferable. Finally, Franke et al [11] used interviews of analyst to understand their information needs with respect to threat actors when handling alerts. We are not familiar with any user studies addressing the effect XAI provides for security analyst when analysing and handling generated alerts.

## III. Experimental setup

A signature-based alert was created as a baseline, and a ML-based alert system was created, with LIME and SHAP used to explain the generated alerts. This was then used in a semi-structured interview with ten security analysts working in a SOC. The following section explains this experimental setup, while §IV summarises the results from the interviews.

### A. Signature-based alert

Suricata, an open source and widely used signature-based detection engine, was used to generate the signature-based alert. The PCAP processing capability of Suricata was used, by generating a custom rule to detect a simple TCP SYN scan reconnaissance attack from a tool called `nmap`:

```
alert tcp any any -> any any (
   msg:"Reconnaissance: nmap 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;)
```

The signature looks for the *S* flag in a TCP packet, while counting the number of unique source IPs. If one IP sends more than 50 packets during one second, an alert is generated. Such alerts will contain a timestamp, signature ID, title, category classification, priority, and relevant systems in the form of IP-addresses and port numbers.

### B. ML-based alert

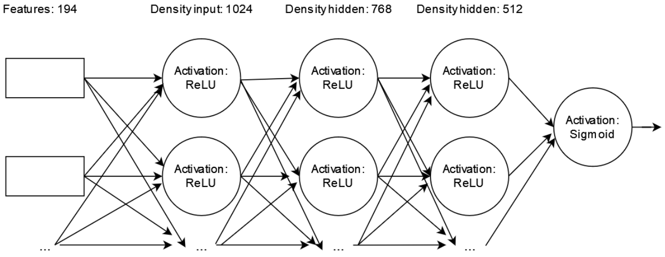

Fig. 3. ML-architecture.

To generate a ML-based alert, we first had to train a ML-model. As the focus is on explainability issues, its performance is not a major issue as long as it is sufficiently realistic and complex. We therefore trained a simple neural network based on [37], as shown in figure 3. Note that 'density' in the figure refers to the number of nodes for each layer. The final sigmoid-activation function will return a number between 0 and 1 where everything above 0.5 is considered malicious.

For the actual training of this ML-model a dataset called UNSW-NB15 [24] was used. Only datapoints related to reconnaissance attacks was included in the training data.

We then used an existing implementation of LIME[4] [29] and a SHAP implementation called DeepSHAP[5] [17]. These provideded local explanation of the predictions as discussed in §II-A.

### C. The interview

Due to the number of potential participants (given our time frame) and, more crucially, to uncover new factors and understand *why* certain opinions were raised, a qualitative approach was taken. To ensure a good portion of exploration, the interview was done in a semi-structured fashion following the approach and guidelines of Elstad [3].

A total of ten people participated in the interview. All had a direct or supporting role in a Security Operations Center (SOC). Most had a master's degree, and the self-reported experience of their work was on average high. Each interview started with a "warm-up" exercise discussing what, in their view, constituted a *good* or *bad* alert. This was followed by a case showcasing two alerts (as described above): one signature-based, and one Machine Learning (ML)-based using no XAI, LIME and SHAP, respectively.

The following signature-based alert was provided:

```
03/09/2022-20:37:49.833584 [1:2300000:3]
Reconnaissance with nmap's SYN SCAN
[Classification:Attempted Information Leak]
```

[4] https://github.com/marcotcr/lime.

[5] https://github.com/slundberg/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 |

Fig. 4. SHAP-scores provided to interviewees.

```
[Priority: 10] {TCP}
10.0.0.99:60522 -> 10.0.0.100:1309
```

The participants were also given the signature in which the alert was generated from (see §III-B). A similar alert was provided for the ML-generated case. For LIME and SHAP, a table with the three features with the highest scores was given. This is illustrated for SHAP in figure 4. A second table, summarising datapoints from the training data was also provided, as was a textual explanation of feature importance and how they might be influential based on the tables.

The questions and discussions in the interviews focused on how the methods differed. Our intention behind this structure was that by focusing on their perceptions of the concrete alerts, we would be able to better elicit their alert requirements.

We refer to [8] for additional details about how the interview was conducted (including the interview guide).

## IV. Results

Signature-based alerts were perceived as detailed, easy to understand and providing a suitably isolated view of the underlying data by the participants in the interview. This was rather unsurprising given most of the participants were used to signature-based alerts. They did however find them to be strict in their capabilities.

ML-based alerts provided a better overall overview of the data and a more precise evaluation, as the influence of features previously considered uninteresting could now be seen. However, this type of alerts was not seen to provide sufficient explanation and control for the actual decision making, and it was unclear how analysts could influence and change future predictions. At a high-level, the participants found both signature- and ML-based alerts to provide some level of alert interpretability and could not see an advantage of either alerts with regards to analysis time. ML-based alerts were however seen to require more knowledge and analytical skills – one reason for this being the higher number of variations (features) provided.

Two rather surprising results was (a) the new insight provided by XAI and (b) a need for global explanation. With respect to (a), one advantage of ML-based alerts (in particular when used for anomaly detection) was the possibility of detecting unseen attacks. Another advantage, highlighted by one participant, was the novel insight of the data used for detection: "*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*". For example, the "*No next header for IPv6*" shown in figure 4 was considered important. Thus, it may also act as a noise filter.

We see (b) as part of a larger question of how to close the "semantic gap" between data science and security analysts. At one end of the spectrum, the argument is to educate security analysts or employing data scientists in the SOC, probably resulting in some change to the processes in the SOC. This is proposed in both [26], [38] and [38]. It has also been proposed to include end-users during development in order to ensure "actionable" alerts [2], [6].

Our initial hypothesis was at the other end of the spectrum, where the ambition is to hide ML-details and provide explanations that are adapted to the existing knowledge and processes in a SOC.[6] We have not considered global explanation as relevant, as we consider them more as support for data scientists when developing, debugging and optimising their models. This is thus seen as an implementation detail we did not consider to be relevant for the security analyst. The results from the interview has however showed that many want a more conscious understanding of the ML-models used to generate the prediction score – and even more than SHAP and LIME provided. A similar result was seen in [26], where an analyst lost trust in the system when provided with just the prediction score of the ML model – thus showing the need for explanation. These results have made us revisit our initial hypothesis, where as many details as possible should be hidden, and include underlying details we initially considered exclusively for developing models. Finally, note that we did not experience the same as a previous study with the Situ system [13]. There, they found that the biggest challenge security analysts faced was a change of habit in the analyst's mindset when moving from a signature-based to a ML-based system.

### A. "Good" alerts and alert enrichment

In addition to our main study of XAI, the "warm-up" exercise of the interview was a discussion of what constitutes a "good" alert. The purpose of this discussion was to enable the interviewees to relate their views on this matter to the XAI-specific questions. Secondly, enrichment of alerts with relevant contextual information was known to us prior to this project: an appropriate alert enrichment is likely to reduce the burden of initial alert triage in the SOC. However, the type of enrichment needed was less clear. Whilst not XAI specific, we did consider this topic to be sufficiently relevant and important to be included in the interview and we briefly summarise the results here.

First and foremost, a good alert must be actionable and trigger a process, or "*a good alert is relevant, meaning that you can act on it*", as was one of the opening statements.

[6] An analogy can be found for high-level programming languages: such programs are often complied to lower-level languages such as C. Here you want the debugger to work in the high-level language and not the compiled C code.

The source of the alert, and what made it trigger, was also considered to be important. The source could for instance be the malware or (Mitre ATT&CK)[7] technique that the model/signature tries to detect. One of the interviewees said "*... if it is C2 we are looking at, then it should be clear in the title.*"[8]. Some sort of example-based explanation was also sought as it is easier for analysts to compare, rather than making up their own attack behavior. An example could be a simulated scenario or data from the training set if feasible.

The source of the data, sensors capturing the data and the underlying infrastructure were all seen as important contextual information by the analyst. It should be clear: what is internal and what is external traffic; which services are running; what the SOC should protect; and information about which parts of the infrastructure belong to the enterprise which the SOC is protecting. The placement of sensors (e.g. if they are behind a firewall, gateway, NAT, etc) was also considered important as this shows which type of traffic the sensors can see and capture.

Enrichment with relevant historical information was also highlighted in the interviews, and providing previous analysis of the same alert type would arguably speed up the analysis. Historical precision of the signature, or, in the case of ML, historical performance[9] of the ML-model that generated the alert was considered relevant. Trust in the supply-chain did also come up in the discussions; for example, if a signature or ML-model is obtained from a stakeholder known for their high quality products, or has a high degree of trust, then this information should be provided. Similarly, if this has been developed internally then such details should also be given. Previous analysis arising from the signature/model should also be used to develop a notion of risk for the alert[10], including the likelihood and consequence of an associated vulnerability or incident.

Enrichment should use both internal and external knowledge as well as threat intelligence to enrich the alerts. Such enrichment should include information related to artefacts, such as IP-addresses and domain names extracted from the logs (data point(s)) in which the alert was generated. Information considered relevant included previous usage such as: previous observations and their location, relevant campaigns or previous incidents, association with known CVEs[11], vulnerabilities, (type of) threat actor, who has information on them, and links to the (US) national vulnerability database[12] which contains information on severity score and weakness enumeration[13]. Finally, information of whether or not a Proof of Concept (POC)[14] has been published should be provided.

## V. Conclusion and future work

As far we are aware, this is the first user study on the impact of XAI to explain alerts to analysts in a SOC.[15] We see this as an important first step towards operationalising XAI for alerts, which will only be possible when understanding the needs of the analysts that will be using it.

Techniques such as SHAP and LIME seems promising, but needs to be further tailored and enriched with contextual information. The interviews have also improved our understanding of the type of information a "good" alert should provide and stressed the importance that the explanations are precise and deemed trustworthy. We did also see the need to include information about the underlying ML-model we originally considered less relevant.

This paper has only documented the first step of a larger research vision with a limited scope. We have only focused on a single tactic of Mitre ATT&CK (Reconnaissance), and all interviewees were from the same enterprise/SOC which may introduce enterprise-specific bias.

Our discussion is limited to SHAP and LIME, but as indicated in figure 1, there are many other XAI-methods to be considered. Examples include model-specific methods (e.g. [35]), counter-factual explanations [23], intrinsic methods like decision trees and the use of knowledge graphs [16].

Finally, is it possible to develop metrics to quantitatively measure and compare ML-models based on how explainable the alerts they generate are? There are proposals for general XAI metrics (e.g. [30]), which can be build upon, however as eluded to in definition 1, explanations must be tailored to their context, which the metrics also need to reflect.

*Acknowledgements*

Thanks to Ann-Kristin Elstad for support with structuring the interview and the anonymous SOC analysts participating in the interview.

## References

[1] 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*, pages 625–633, May 2007. ISSN: 0743-166X.

[2] 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, pages 1–9, New York, NY, USA, August 2018. Association for Computing Machinery.

[3] Ann-Kristin Elstad. *Critical Success Factors When Implementing an Enterprise System - An Employee Perspective*. PhD thesis, Norges Handelshøyskole, 2014.

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, June 2020.

[5] Bushra A. Alahmadi and Louise Axon. 99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms. In *Proceedings of the 31st USENIX Security Symposium*, 2022.

[7] https://attack.mitre.org/.

[8] Here, 'title' refers to the name of the signature and for ML-models this could be the type of tactic or technique the model was trained on.

[9] E.g. metrics such as accuracy, precision, recall and $F_\beta$.

[10] This has some similarities with Splunk's risk-based alerting; see e.g. https://splk.it/3ycCQgS

[11] https://www.cve.org/

[12] https://nvd.nist.gov.

[13] http://cwe.mitre.org.

[14] A POC is a publicly available working exploit.

[15] Franke et al. [11] also points to both the need and lack of user studies of XAI in this setting.

[6] Dylan Cashman, Shah Rukh Humayoun, Florian Heimerl, Kendall Park, Subhajit Das, John Thompson, Bahador Saket, Abigail Mosca, John Stasko, Alex Endert, Michael Gleicher, and Remco Chang. A User-based Visual Analytics Workflow for Exploratory Model Analysis. *Computer Graphics Forum*, 38(3):185–199, June 2019. Eurographics Conference on Visualization (EuroVis) 2019.

[7] Shuchisnigdha Deb and David Claudio. Alarm fatigue and its influence on staff performance. *IIE Transactions on Healthcare Systems Engineering*, 5(3):183–196, July 2015. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/19488300.2015.1062065.

[8] Håkon Svee Eriksson. A user-centric approach to explainable ai in a security operation center environment. Master thesis, University of Oslo, 2022. Available from: https://www.duo.uio.no/handle/10852/96758.

[9] 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)*, pages 173–175, July 2017.

[10] Erik M. Ferragut, David M. Darmon, Craig A. Shue, and Stephen Kelley. Automatic construction of anomaly detectors from graphical models. In *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pages 9–16, April 2011.

[11] Ulrik Franke, Annika Andreasson, Henrik Artman, Joel Brynielsson, Stefan Varga, and Niklas Vilhelm. Chapter 10 - cyber situational awareness issues and challenges. In *Cybersecurity and Cognitive Science*, pages 235–265. Academic Press, 2022.

[12] S. García, M. Grill, J. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, September 2014.

[13] John R. Goodall, Eric D. Ragan, Chad A. Steed, Joel W. Reed, G. David Richardson, Kelly M.T. Huffer, Robert A. Bridges, and Jason A. Laska. Situ: Identifying and Explaining Suspicious Behavior in Networks. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):204–214, January 2019. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[14] Balázs Péter Hámornik and Csaba Krasznay. A Team-Level Perspective of Human Factors in Cyber Security: Security Operations Centers. In Denise Nicholson, editor, *Advances in Human Factors in Cybersecurity*, Advances in Intelligent Systems and Computing, pages 224–236, Cham, 2018. Springer International Publishing.

[15] Muhammad Usama Islam, Md. Mozaharul Mottalib, Mehedi Hassan, Zubair Ibne Alam, S. M. Zobaed, and Md. Fazle Rabby. The Past, Present, and Prospective Future of XAI: A Comprehensive Review. In *Explainable Artificial Intelligence for Cyber Security: Next Generation Artificial Intelligence*, Studies in Computational Intelligence, pages 1–29. Springer International Publishing, Cham, 2022.

[16] Jian-hua Li. Cyber security meets artificial intelligence: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(12):1462–1474, December 2018.

[17] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[18] Shraddha Mane and Dattaraj Rao. Explaining Network Intrusion Detection System Using Explainable AI Framework. *arXiv:2103.07110 [cs]*, March 2021. arXiv: 2103.07110.

[19] Manu Joseph. Interpretability part 3: opening the black box with LIME and SHAP. Section: 2019 Dec Tutorials, Overviews.

[20] Sherin Mary Mathews. Explainable Artificial Intelligence Applications in NLP, Biomedical, and Malware Classification: A Literature Review. In Kohei Arai, Rahul Bhatia, and Supriya Kapoor, editors, *Intelligent Computing*, Advances in Intelligent Systems and Computing, pages 1269–1292, Cham, 2019. Springer International Publishing.

[21] Natalia Miloslavskaya. Analysis of SIEM Systems and Their Usage in Security Operations and Security Intelligence Centers. In Alexei V. Samsonovich and Valentin V. Klimov, editors, *Biologically Inspired Cognitive Architectures (BICA) for Young Scientists*, Advances in Intelligent Systems and Computing, pages 282–288, Cham, 2018. Springer International Publishing.

[22] Christoph Molnar. *Interpretable Machine Learning*. Molnar, Christoph, 2022.

[23] 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*, pages 607–617, January 2020.

[24] 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)*, pages 1–6, 2015.

[25] Megan M. Nyre-Yu. *Determining System Requirements for Human-Machine Integration in Cyber Security Incident Response*. thesis, Purdue University Graduate School, October 2019.

[26] Sean Oesch, Robert Bridges, Jared Smith, Justin Beaver, John Goodall, Kelly Huffer, Craig Miles, and Dan Scofield. An Assessment of the Usability of Machine Learning Based Tools for the Security Operations Center. *arXiv:2012.09013 [cs]*, December 2020.

[27] Jose N. Paredes, Juan Carlos L. Teze, Gerardo I. Simari, and Maria Vanina Martinez. On the Importance of Domain-specific Explanations in AI-based Cybersecurity Systems (Technical Report). Technical Report arXiv:2108.02006, arXiv, August 2021.

[28] A. Rawal, J. Mccoy, D. B. Rawat, B. Sadler, and R. Amant. Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges and Perspectives. *IEEE Transactions on Artificial Intelligence*, 1(01):1–1, December 5555. Place: Los Alamitos, CA, USA Publisher: IEEE Computer Society.

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv:1602.04938 [cs, stat]*, August 2016.

[30] Avi Rosenfeld. Better metrics for evaluating explainable artificial intelligence. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems*, pages 45–50, 2021.

[31] Lloyd S. Shapley. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952.

[32] Michael R. Smith, Nicholas T. Johnson, Joe B. Ingram, Armida J. Carbajal, Bridget I. Haus, Eva Domschot, Ramyaa Ramyaa, Christopher C. Lamb, Stephen J. Verzi, and W. Philip Kegelmeyer. Mind the gap: On bridging the semantic gap between machine learning and malware analysis. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*, AISec'20, page 49–60, New York, NY, USA, 2020. Association for Computing Machinery.

[33] R. Sommer and V. Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316, May 2010. ISSN: 2375-1207.

[34] Hatma Suryotrisongko, Yasuo Musashi, Akio Tsuneda, and Kenichi Sugitani. Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing. *IEEE Access*, 10:34613–34624, 2022. Conference Name: IEEE Access.

[35] Gabriel Terejanu, Jawad Chowdhury, Rezaur Rashid, and Asif Chowdhury. Explainable Deep Modeling of Tabular Data using TableGraphNet. Technical Report arXiv:2002.05205, arXiv, February 2020.

[36] Manfred Vielberth, Fabian Böhm, Ines Fichtinger, and Günther Pernul. Security Operations Center: A Systematic Study and Open Challenges. *IEEE Access*, 8:227756–227779, 2020. Conference Name: IEEE Access.

[37] Rahul Vigneswaran. Intrusion Detection Systems, May 2022. original-date: 2018-09-22T09:24:43Z. Accessible: https://github.com/rahulvigneswaran/Intrusion-Detection-Systems.

[38] Maonan Wang, Kangfeng Zheng, Yanqing Yang, and Xiujuan Wang. An Explainable Machine Learning Framework for Intrusion Detection Systems. *IEEE Access*, 8:73127–73141, 2020. Conference Name: IEEE Access.

[39] Łukasz Wawrowski, Marcin Michalak, Andrzej Białas, Rafał Kurianowicz, Marek Sikora, Mariusz Uchroński, and Adrian Kajzer. Detecting anomalies and attacks in network traffic monitoring with classification methods and XAI-based explainability. *Procedia Computer Science*, 192:2259–2268, January 2021.