IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 24 August 2022, accepted 17 October 2022, date of publication 25 October 2022, date of current version 31 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3216617

SURVEY

Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities

SUBASH NEUPANE[1], JESSE ABLES[1], (Graduate Student Member, IEEE), WILLIAM ANDERSON[1], SUDIP MITTAL[1], (Member, IEEE), SHAHRAM RAHIMI[1], (Member, IEEE), IOANA BANICESCU[1], (Life Senior Member, IEEE), AND MARIA SEALE[2]

[1]Department of Computer Science and Engineering, Mississippi State University, Mississippi, MS 39762, USA
[2]U.S. Army Engineer Research and Development Center, Vicksburg, Mississippi, MS 39180, USA

Corresponding author: Subash Neupane (sn922@msstate.edu)

This work was supported by Mississippi State University through the U.S. Department of Defense (DoD) High Performance Computing Modernization Program through the U.S. Army Engineering Research and Develop Center (ERDC) under Grant W912HZ-21-C0058.

ABSTRACT The application of Artificial Intelligence (AI) and Machine Learning (ML) to cybersecurity challenges has gained traction in industry and academia, partially as a result of widespread malware attacks on critical systems such as cloud infrastructures and government institutions. Intrusion Detection Systems (IDS), using some forms of AI, have received widespread adoption due to their ability to handle vast amounts of data with a high prediction accuracy. These systems are hosted in the organizational Cyber Security Operation Center (CSoC) as a defense tool to monitor and detect malicious network flow that would otherwise impact the Confidentiality, Integrity, and Availability (CIA). CSoC analysts rely on these systems to make decisions about the detected threats. However, IDSs designed using Deep Learning (DL) techniques are often treated as black box models and do not provide a justification for their predictions. This creates a barrier for CSoC analysts, as they are unable to improve their decisions based on the model's predictions. One solution to this problem is to design *explainable IDS* (X-IDS). This survey reviews the state-of-the-art in explainable AI (XAI) for IDS, its current challenges, and discusses how these challenges span to the design of an X-IDS. In particular, we discuss black box and white box approaches comprehensively. We also present the tradeoff between these approaches in terms of their performance and ability to produce explanations. Furthermore, we propose a generic architecture that considers human-in-the-loop which can be used as a guideline when designing an X-IDS. Research recommendations are given from three critical viewpoints: the need to define explainability for IDS, the need to create explanations tailored to various stakeholders, and the need to design metrics to evaluate explanations.

INDEX TERMS Explainable intrusion detection systems, explainable artificial intelligence, machine learning, deep learning, white box, black box, explainability, cybersecurity.

I. INTRODUCTION AND MOTIVATION

The use of Artificial Intelligence (AI) and Machine Learning (ML) to solve cybersecurity problems has been gaining traction within industry and academia, partly as a response to widespread malware attacks on critical systems, such as cloud infrastructures, government institutions, etc. [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. AI- and ML-assisted cybersecurity offers data-driven automation that could enable security systems to identify and respond to cyber threats in real time. Many of these AI-based cyber defense systems are hosted in an organizational Cyber Security Operations Center (CSoC). CSoCs operated by security analysts act as a cybersecurity information hub and a defense base. Here the task is to orchestrate different security systems that are a part of an organization's overall cybersecurity framework,

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed.

many of which have AI components. Examples of these security systems include Security Information and Event Management (SIEM) systems, vulnerability assessment solutions, governance, risk and compliance systems, application and database scanners, Intrusion Detection Systems (IDS), user and entity behavior analytics, Endpoint Detection and Remediation (EDR), etc. Here the security analysts maintain an ''organizational state'', keeping themselves one step ahead of the attackers to prevent potential intrusions [12].

The term ''intrusion detection'' originated in the early 1980s with James Anderson's seminal paper [13]. Dorothy E. Denning [14], following Anderson's work, proposed the first functional IDS in the mid-1980s. An IDS is a software or hardware security system that automates the process of monitoring and analyzing events occurring within a computer system or network for indications of potential security problems before they inflict widespread damage [15], [16].

In general, an intrusion results in a breach of at least one of the principles: *Confidentiality*, *Integrity*, or *Availability* (CIA). These tenets of security are used when protecting modern data infrastructure. They refer to the permissions to access or modify data, the prevention of improper data modification, and the ability to access data. The objective of an IDS is to detect misuse, unauthorized use (outsider without authorization), and abuse (abusing privilege-e.g., insider threat) within an organization, and much research has been done to improve the operational capacity of these IDS [17], [18], and [19].

The literature shows that numerous IDSs have been developed through the application of a variety of techniques from an array of disciplines, including statistical methods, ML techniques, and others [20]. At present, ML and Deep Learning (DL) techniques are widely used to develop IDS because of their ability to attain a high detection rate [16]. This adoption is also attributed to the fact that IDSs based on ML/DL techniques are much more efficient, accurate, and extendable as compared to their counterparts developed using other techniques. The surveys in [1], [21], and [22] primarily focus on intrusion detection techniques based on deep learning. However, *the techniques described in the preceding surveys are deficient in their ability to explain their inference processes and final results*, and they are *frequently treated as a black box by both developers and users* [23]. As a result, there is growing concern about the possibility of bias in these models, which necessitates the requirements for model transparency and post-hoc explainability [24]. Unfortunately, the majority of black box IDS described in the literature are opaque and much is needed to augment transparency.

It is apparent that these opaque/non-transparent models can achieve *impressive prediction accuracies*; however, they lack justification for their predictions. This is due to their nested and non-linear structure, which makes it difficult to identify the precise information in the data that influences their decision-making [25]. Such a *lack of understanding about the inner workings of opaque AI models* or an inability to traverse back from the outputs to the original data raises user trust issues [26]. This black box nature of the models creates problems for several domains in which AI or components of AI are integrated [27]. For example, in the context of an IDS, CSoCs analysts are tasked with the responsibility of analyzing IDS alerts for a variety of purposes, including alert escalation, threat and attack mitigation, intelligence gathering, and forensic analysis among others [28]. The lack of explanation of alerts generated by an IDS creates a barrier for task analysis and subsequently impedes decision-making.

In addition to the issues of *transparency* and *trust* surrounding AI systems, there exists yet another issue referred to as the problem of *decomposability*, specifically for systems built with DL models (See Section IV-A3 for IDS based on the decomposition approach). AI systems that are designed using DL techniques are difficult to interpret due to their inability to be decomposed into intuitive components [29]. The difficulty in interpreting DL models jeopardizes their actual use in production, as the computation behind their decisions are unknown [11]. *Explainable AI* (XAI) seeks to remedy this and other problems.

According to the Defense Advanced Research Projects Agency (DARPA), XAI systems are able to explain their reasoning to a human user, characterize their strengths and weaknesses, and convey a sense of their future behavior [24]. In this sense, by justifying specific decisions, XAI systems aid users in comprehending the model and assisting them in maintaining and effectively using it. On the other hand, transparency about predictions contributes to the development of trust in a system's intended behavior and provides users with confidence that they are performing tasks correctly.

Transparency is an open problem in the field of intrusion detection. Cybersecurity professionals now frequently make decisions based on the recommendations of an AI-enabled IDS. Therefore, the predictions made by the model should be understandable [11]. For instance, when an IDS model is presented with zero-day attacks, the model may misclassify the attacks as normal, resulting in a system breach. Understanding why specific samples are misclassified is the first step toward debugging and diagnosing the system. It is critical to provide detailed explanations for such misclassifications, so as to determine the appropriate course of action to prevent future attacks [6]. Therefore, an IDS should go beyond merely detecting intrusions-i.e., it should provide reasoning for the detected threat. The explanations in the form of correlations of various factors (for example, time of intrusion, type, suspicious network flow) influencing the predicted outcome can assist cybersecurity analysts in quickly analyzing tasks and making decisions [28].

The goal of *XAI in the field of intrusion detection* is to build operator trust and allow for more control of autonomous AI subsystems. Explainable Intrusion Detection Systems (X-IDS) will help build trust in these systems while also aiding CSoC analysts in their task of defending systems.

The major contributions of the paper are as follows:

- We present the state-of-the-art of the XAI approach and discuss the critical issues that surround it, most importantly, how these issues relate to the intrusion detection domain. We propose a taxonomy based on a literature review to help lay the groundwork for formally defining explainability in intrusion detection.

- A comprehensive survey of the current landscape of X-IDS implementations is presented, with an emphasis on two major approaches: black box and white box. The distinction between the two approaches is discussed in detail, as is the rationale for why the black box approach with post-hoc explainability is more appropriate for intrusion detection tasks.
- We propose a generic explainable architecture with a user-centric approach for designing X-IDS that can accommodate a wide variety of scenarios and applications without adhering to a specific specification or technological solution.
- We discuss the challenges inherent in designing X-IDS and make research recommendations aimed at effectively mitigating these challenges for future researchers interested in developing X-IDS.

The remainder of this paper is organized as follows. Section II provides the background on explainable artificial intelligence (XAI). Section III summarizes our survey and taxonomy. Following that, in Section IV, we review the literature on black box and white box X-IDS approaches. Section V introduces a generic X-IDS architecture that future researchers can use as a guide. Section VI identifies research challenges and makes recommendations to future researchers. Finally, Section VII concludes this survey.

II. EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI)

The definition of what constitutes an explanation in AI remains an open research question. In the available literature, there are various definitions of the 'explainable AI' (XAI). Lent et al. [30] defines XAI as a system that *"provides an easily understandable chain of reasoning from the user's order, through the system's knowledge and inference, to the resulting behavior"*. The authors of this work used the term XAI to describe their system's ability to explain the behavior of AI-controlled entities. However, the recent definition by DARPA [31] indicates XAI as the intersection of different areas including machine learning, human computer interface, and end user explanation.

XAI has the potential to offer significant benefits to a broad range of domains that rely on artificial intelligence systems. Most importantly, the impact of XAI in the decision-making process for stakeholders in the IDS ecosystem is considerable. IDSs based on deep learning models can be more efficient in detecting malicious traffic with high accuracy. However, a CSoC analyst is still left with a significant task, i.e., to determine why the flow is malicious, and how to best deal with the attack. One solution to this problem is the post-hoc explainability offered by XAI. In [11], for example, the authors demonstrate how the SHapley Additive exPlanation (SHAP) framework [32] can be utilized to gain a deeper understanding of the model characteristics that contribute the most to various types of attacks. Similarly, misclassification is another important issue within the IDS ecosystem. Misclassification of malicious network flow as benign could be catastrophic to an organization (CSoC). For example, the authors in [6], demonstrated how the counterfactual technique can be used to explain misclassification. User trust in IDS predictions is imperative, and providing justification is as important as the prediction itself. For example, in [33], the authors illustrated how user trust can be garnered by providing input feature relevance scores (Layer-wise Relevance Propagation (LRP) method) that indicate the contribution of each input feature to the detection of the intrusion.

Apart from IDS systems, currently, XAI is being used in mission-critical systems and defense [30], [31]. To foster the trust of AI systems in the transportation domain, researchers are proposing explanations systems [34]. Some works based on image processing with explainability is found in [35], [36], [37], and [38]. Transparency regarding decision-making processes is critical in the criminal justice system [27], [39]. Various explainable methods for judicial decision support systems have been proposed by authors in [40], [41], and [42]. Model explainability is essential for gaining trust and acceptance of AI systems in high-stakes areas, such as healthcare, where reliability and safety are critical [43], [44]. Medical anomaly detection [45], healthcare risk prediction system [46], [47], [48], [49], genetics [50], [51], and healthcare image processing [52], [53], [54] are some of the areas that are moving towards adoption of XAI. Another area is finance, such as AI-based credit score decisions [55], [56] and counterfeit banknotes detection [57]. Support for XAI in academia for evaluation tasks are found in [58], [59], and [60]. Lastly, in the entertainment industry XAI for recommender systems is found in the works of [61], [62], and [63].

Arrieta et al. [64] argue that one of the issues that hinders the establishment of common ground for the meaning of the term 'explainability' in the context of AI is the interchangeable misuse of 'interpretability' and 'explainability' in the literature. Interpretability is the ability to explain or convey meaning in human-comprehensible terms [64]. This translates into the ability of a human to understand the model's reasoning without the need for additional explanations [65]. On the other hand, explainability is associated with the concept of explanation as a means of interface between humans and a decision-maker (model) that is both accurate and comprehensible to humans [65]. In this sense, if system users need an explanation as a proxy system to understand the reasoning process, that explanation is precisely represented by the XAI.

A central concept that emerges from all the preceding definitions of the XAI is 'understandability', which is the degree to which a human can comprehend a decision made with respect to a model. However, understandability is tightly coupled with the characteristics of the system's users. For instance, whether or not the explanation made the concept clear or simple to understand is entirely dependent on the audience.

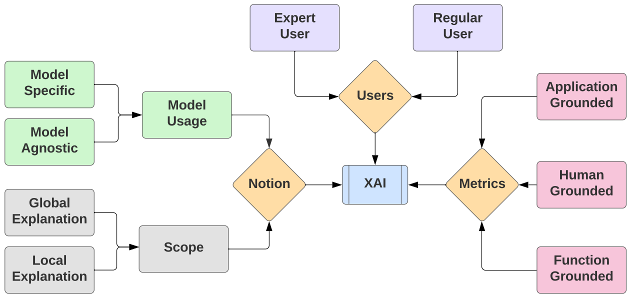

FIGURE 1. A taxonomic approach to explainability definition based on explainability concepts, formalizing explainability tasks from the standpoint of stakeholders, and evaluating explainability techniques. Green represents model dependency, while grey represents the scope of the explanations. Light purple represents various types of stakeholders in the IDS ecosystem. Pink represents techniques to evaluate explanations.

Despite the widespread recognition of the importance of explainability, researchers are struggling to establish universal, objective criteria for developing and validating explanations [66]. This is because XAI is plagued by inherent challenges that need addressing to foster its development. These include (i) achieving consensus on the right notion of model explainability, (ii) identifying and formalizing explainability tasks from the perspectives of various stakeholders, and (iii) designing measures for evaluating explainability techniques [67].

To address these challenges, we propose a taxonomy as depicted in Figure 1 based on our literature review to lay the groundwork for formally developing and validating explanations. In the following subsections, we describe in detail the concepts related to XAI including its notions, its meaning to various stakeholders of the system, and metrics to evaluate explanations.

A. NOTIONS OF EXPLAINABILITY

Several approaches to explanation methods have been proposed by different authors in the pursuit of explaining AI systems. The authors in [65] conducted a survey of black box specific explainability methods and proposed a taxonomy for XAI systems based on four characteristics: (i) the nature of the problem; (ii) the type of explainer used; (iii) the type of black box model processed by the explainer; and (iv) the type of data supported by the black box.

In another work [68], the authors presented notions related to the concept of explainability in two clusters. The first cluster refers to attributes of explainability – it contains criteria and characteristics used by scholars in trying to define the construct of explainability. The second cluster refers to the theoretical approaches for structuring explanations. Das and Rad in [69] proposed a taxonomy for categorizing XAI techniques based on explanation scope, algorithm methodology, and usage. Similarly, the authors in [55] surveyed over 180 articles related to explainability and categorized explainability using three criteria: the complexity of interpretability, the scoop of interpretability, and model dependency. The first criterion emphasizes the difficulty of interpreting and explaining complex models, such as those based on deep learning. The second criterion differentiates between local and global explanations, while the third criterion discusses model-specific and model-agnostic explanations. On the other hand, Pantelis et al. in [70] divided explainability methods into four groups based on: the data types used, the scope of explanation, the purpose of explanation, and the model usage.

A common category found in the literature regarding the taxonomy of explainability is the *scope of explainability* and *model dependency*. The following subsections describe these categories in greater detail.

1) LOCAL EXPLAINABILITY

The ability to explain a single prediction or decision is an example of local explainability. This explainability is used to generate a unique explanation or justification of the specific decision made by the model [55].Some of the local explanation methods include the Local Interpretable Model Agnostic Explanation (LIME) [71], the Anchors [72] and the Leave One Covariate Out (LOCO) [73]. LIME was originally proposed by Ribeiro et al. [71] who used a surrogate model to approximate the predictions of the black box model. Rather than training a global surrogate model, LIME uses a local surrogate model to interpret individual predictions.

To explain the behavior of complex models with high precision rules called *Anchors*, representing local, *sufficient conditions* for predictions, the same authors proposed an extension to the LIME method in [72]. Another popular technique for generating local explanation models with local variable importance measures is LOCO [73].

Lundberg et al. [32] proposed a game-theoretic optimal solution based on Shapley values for model explainability referred to as Shapely Additive Explanations (SHAP). SHAP calculates the significance of each feature in each prediction. The authors have demonstrated the equivalence of this model among various local interpretable models including LIME [71], Deep Learning Important FeaTures (DeepLIFT) [74], and LayerWise Relevance Propagation (LRP) [75]. The SHAP value can be computed for any model, not just simple linear models.

2) GLOBAL EXPLAINABILITY

The global explainability of a model makes it easier to follow the reasoning behind all the possible outcomes. These models shed light on the model's decision-making process as a whole, resulting in an understanding of the attributions for a variety of input data [69].

The LIME [71] model was extended with a 'submodular pick algorithm' (SP-LIME) in order to comprehend the model's global correlations. By providing a non-redundant global decision boundary for the machine learning model, LIME provides a global understanding of the model from individual data instances using a submodular pick algorithm.

Concept Activation Vectors (CAVs) proposed by Kim et al. [76] is another global explainability method. This model can interpret the internal states of a neural network in the human-friendly concept domain. In another work, Yang et al. [77] proposed a novel method, the Global Interpretation via Recursive Partitioning (GIRP), to construct a global interpretation tree based on local explanations for a variety of machine learning models. Other methods of global explanation include an explanation by information extraction [78]. In this study, the authors propose a method of information extraction that is only lightly supervised and provides a global interpretation. They demonstrated that interpretable models can be generated when representation learning is combined with traditional pattern-based bootstrapping.

3) MODEL-SPECIFIC INTERPRETABILITY

The use of model-specific interpretability methods is restricted to a limited number of model classes. With these methods, we are restricted to using only models that provide a specific type of interpretation, which can reduce our options for using more accurate and representative models.

4) MODEL-AGNOSTIC INTERPRETABILITY

Methods that are model agnostic are not tied to any specific type of ML model, and are by definition modular, in the sense that the explanatory module is unrelated to the model for which it generates explanations. Model-agnostic interpretations are used to interpret artificial neural networks (ANNs) and can be local or global. In their survey [69], the authors argue that a significant amount of research in XAI is concentrated on model-agnostic post-hoc explainability algorithms, due to their ease of integration and breadth of application. Based on other reviewed papers, the authors [55] broadly categorize the techniques of model-agnostic interpretability into four types, including visualization, knowledge extraction, influence methods, and example-based explanations.

B. FORMALIZING EXPLAINABILITY TASKS FROM THE USER PERSPECTIVES

To be explainable, a machine learning model must be human-comprehensible. This presents a challenge for the development of XAI because it entails communicating a complex computational process to humans. The interpretable element that serves as the foundation of explanation is highly dependent on the question of *"who"* will receive the explanation. Rosenfeld et al. [79] identified three targets of explanation, including regular user, expert user, and the external entity. According to the authors, an explanation should be specific to user types. For instance, in a legal scenario, the explanation must be made to the expert users, not the regular users. On the other hand, if explanations are geared towards regular users, then the chance of developing trust and acceptance of XAI methods is high. To address the issue of stakeholder-specific explanation requirements, IBM developed an open-source toolkit known as AI Explainability 360 (AIX360) [80].

Adadi et al. [55] emphasize the significance of humans-in-the-loop approach for explainable systems from two perspectives: Human-like explanation and Human-friendly explanation. The first aspect focuses on how to produce explanations that simulate the human cognitive process, while the second aspect is concerned with developing explanations that are centered on humans.

Section V discusses the importance of human-centered design when developing X-IDS systems, and Section VI-B examines the explainability requirements imposed by various stakeholders in the IDS ecosystem.

C. MEASURES FOR EVALUATING EXPLAINABILITY TECHNIQUES

There have been few studies on evaluating explanations and quantifying their relevance despite the growing body of research that produces explainable ML methods. Doshi-Velez and Kim [81] proposed the three classes as evaluation methods for interpretability, including application-grounded, human-grounded, and functionally grounded methods. Application-grounded evaluation is concerned with the impact of the interpretation process's results on the human, domain expert, or end user, in terms of a well-defined task or application. Human-grounded evaluation is concerned with conducting simplified application-grounded evaluation where experiments are run with regular users rather than domain experts. Functionally grounded evaluation does not require human subjects, and rather uses formal, well-defined mathematical definitions of interpretability to determine the method's quality.

On the other hand, in [82], the authors outline three different evaluation criteria of explanations for deep networks, such as processing, representation, and explanation producing. The first criterion includes techniques that simulate data processing to generate insights about the relationships between a model's inputs and outputs. The second criterion describes an approach on how data is represented in networks and explains the representation. The third criterion states that the explanation-producing systems can be evaluated according to how well they match user expectations.

To evaluate local explanation, IBM in their toolkit AIX360 [80] suggested two metrics such as Faithfulness [83] and Monotonicity [84], to quantify the "goodness" of a feature-based local explanation. Other types of evaluation criteria found in the body of literature include completeness compared to the original model, ability to detect models with biases, completeness as measured on a substitute task, and human evaluation.

Another significant piece of work that could serve as a benchmark for evaluating explanations is the Florida Institute for Human and Machine Cognition's psychological model of explanation (IHMC) [24]. Section VI-C provides greater detail about this proposed model.

Next, we describe our survey approach and develop a taxonomy for X-IDS grounded in the current literature.

III. SURVEY AND TAXONOMY

The term *intrusion* refers to any unauthorized activity occurring within a network or system. An IDS is a collec-

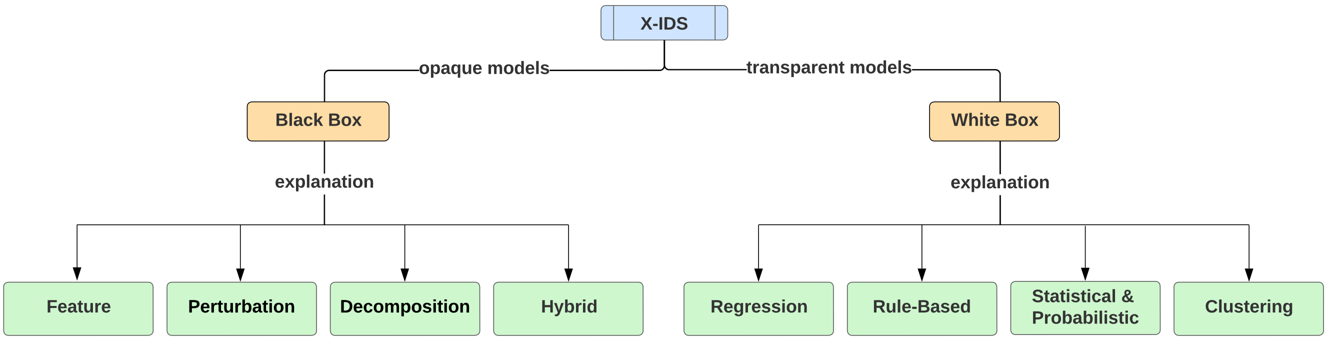

FIGURE 2. An overview of our proposed taxonomy. We categorize X-IDS techniques into two families, white box and black box. White box approaches encompass the techniques of Regression, Rule-Based, Clustering, and Statistical & Probabilistic Methods. Black box approaches encompass Feature, Perturbation, Decomposition, and Hybrid approaches. These approaches define the method of explainability to interpret the model's decision process.

tion of tools, methods, and resources that assist CSoC analysts in identifying, assessing, and reporting intrusions. Intrusion detection is typically a component of protection that surrounds a system and is not a stand-alone protection measure [85]. IDS are classified according to where they look for intrusive behavior: *host-based* or *network-based*. A host-based IDS monitors traffic that is originating from and coming to a specific host. Network-based IDS are strategically positioned in a network to analyze incoming and outgoing communication between network nodes.

IDS are categorized based on three detection techniques: *signature-based*, *anomaly-based*, and *hybrid*. Signature-based IDS monitors network traffic and compares it to a database of known malicious threats' signatures or attributes. However, they are incapable of detecting zero-day attacks, metamorphic threats, or polymorphic threats [86]. On the other hand, anomaly-based IDS look for patterns in data that do not conform to expected behavior [87], allowing them to recognize such threats. However, these detection systems are susceptible to higher false positive rates because they may categorize previously unseen, yet legitimate, system behaviors as anomalies [19]. Hybrid IDS integrate both signature-based and anomaly-based detection methods, which allows for an increased detection rate of known intrusions, the ability to detect unseen intrusions, and reduce false positives.

As previously stated, prior work has focused on XAI from the lens of explainability, qualifying the definitions of *notion*, *users*, and *metrics* (See Section II). This survey follows that direction by creating a taxonomy surrounding current XAI techniques for IDS. The focus is on their relevance and applicability to the domain of intrusion detection, with a particular emphasis on the current hierarchy of families, strengths and weaknesses, and any challenges or assumptions that come with their application. A summary of our taxonomy can be seen in Figure 2. The two primary families of XAI techniques are those of white box models and black box models which greatly affect our survey taxonomy for approaches to X-IDS. The survey of existing systems based on the taxonomy in Figure 2, is available in Section IV. Next, we describe the salient features of white box models and black box models.

A. SALIENT FEATURES OF WHITE BOX TECHNIQUES

White box models provide results that are easy to understand [88]. This *easy to understand* condition is typically defined as an explainable outcome understood by an expert in the field. In practice, this definition is more associated with the popular suite of machine learning models that existed prior to the rise in popularity of neural network based approaches. White box models, while generally not as efficacious as their black box counterparts, bring a layer of transparency that is intrinsic to their decision process. This trait is often preferred, if not a requirement, in domains where the decision system is sensitive or requires a high degree of auditing. These models cover a wide variety of techniques that fall into four distinct families: *Regression*, *Rule-Based*, *Clustering*, and *Statistical & Probabilistic Methods*.

Regression-based approaches comprise the family of regression analysis. These approaches have a well formed background of statistical support and maturity. Therefore, these models are most often employed in the early stages of modeling, in the pipelines of more complex models, and in domains where scrutiny and transparency are of paramount importance. Although not a focal point of comparison for this paper, regression models are highly computationally efficient, allowing for rapid construction, as well as deployment into low-resource systems where detection time is critical, such as IoT edge devices. Regression approaches can be split into Parametric Regression and Non-parametric Regression. The former enforces a constraint on model expectation via a restriction on the parameters of the model, making this modeling approach best for when certain assumptions can be met. The latter enforces no such constraint, which decreases overall interpretability but increases the application to a wider variety of data and assumptions. Popular regression techniques are Linear Regression (LR), Logistic Regression (LoR), various non-linear models, Poisson Regression, Kernel Regression (KR), and Spline Smoothing.

Rule-based approaches leverage a learned set of rules as a means of the model decision process, and thusly, model explainability. Rule based explanations are perhaps the most practical, as they mimic the human decision making process when it comes to defining an anomaly. This process also allows learned rules to then be incorporated into Signature-Based IDS (SIDS), allowing Anomaly-Based IDS (AIDS) to serve as zero-day identifiers and rule miners. Rule-based approaches benefit from the allowance of a very tight definition of rules, known as hard rules or crisp rules, or for a relaxed fuzzy-rule based approach, allowing flexibility and further statistical inference to be rendered on them. A popular approach to modeling for rule-based explanations is the Decision Tree and its many variants.

Statistical & Probabilistic Methods is a broad category for the numerous statistical models of reasoning that exist in the literature. Notably, many of these methods have seen a decline in use as a compliment to the rise in popularity of various black box methods. These less frequently used methods are appropriate for application in specific scenarios or in larger pipelines for multi-stage IDS. Examples of such approaches include moment-based approaches, statistical ensembles, Markov Models, Baysian Networks, and others which are covered more specifically in IV-B3.

Clustering-based approaches use supervised or unsupervised learning to aggregate similar data objects. This *similar* condition is defined by a similarity, or dissimilarity, measure. Traditionally these methods are defined on distance based metrics such as Euclidean, Manhattan, Cosine Measure, Pearson coefficient, and many others. Other attempts to define similarity have had success in the graph representation domain, using graph-based clustering algorithms to accomplish this task. Clustering, due to its ability to be leveraged as an unsupervised learner, still retains a high degree of use due to the importance of data mining for intrusion detection. Examples of popular clustering algorithms are K-Means, Self-Organzing Maps (SOMs), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Agglomerative Clustering, and Spectral Clustering.

B. SALIENT FEATURES OF BLACK BOX TECHNIQUES

Black box models are models where the decision systems are considered opaque [65]. These systems, composing nearly all of the state-of-the-art, are limited due to the lacking ability of model inspection and evaluation. Therefore, if these systems are to be utilized in decision sensitive domains, i.e, those whose applications require safety, privacy, and fairness, some degree of exploration and evaluation of their decision process must be possible. Currently, there exists no singular solution to the black box inspection problem. However, many candidate explanations have emerged, exploring and exploiting various aspects of the machine learning process to create explanations for black box models. These candidate explanations currently fall into four distinct families: *Feature*, *Perturbation*, *Decomposition*, and *Hybrid*.

Feature-based explanations target features as the method of explanation. The goal of feature attribution is to determine how much each feature is responsible for the output prediction. Features were one of the first methods of explainability in black box models due to their impact on model performance and human interpretable relevance. Examples of popular feature based explanations are Partial Dependence Plot (PDP) [89], Accumulated Local Effects (ALE) [90], Individual Conditional Expectations (ICE) [91], H-statistic [92], and SHapley Additive exPlanations (SHAP) [32].

Perturbation-based explanations study changes to the output space with perturbations to the input space. Due to this property, perturbation techniques can be deployed to any general input space, such as tabular data, images, or text. In particular, model sensitivity to feature perturbations has long been regarded as a measure of feature importance. Saliency maps, Randomized Input Sampling for Explanation of Black Box Models (RISE) [93], and Local Interpretable Model-Agnostic Explanations (LIME) [71] are popular perturbation methods.

Decomposition-based explanations decompose the original model prediction. Much like the previous two methods, the goal of decomposition is to allocate a measure of importance to the input space; however, this method does so by the decomposition of a model signal, such as the model's gradients. This is predicated on the assumption that large gradients play a role in shaping explanations. However, gradients are not the only method of decomposition. Many approaches exist, such as Gradient * Input [94], Integrated Gradients (IG) [95], Grad-CAM [96], DeepLIFT [74], Deep Taylor Decomposition (DTD) [97] and Layerwise Relevance Propagation (LRP) [75].

Hybrid-based explanations encapsulate a type of model construction often demonstrated in pipelined machine learning architectures. These models can range from ensembles, a blend of white box and black box approaches working in tandem, to carefully composed IDS pipelines encapsulating many of the best state-of-the-art approaches. Therefore, hybrid approaches present the most variability of explanations, with respect to methodology, location of explanations, and application.

Next, we use the taxonomy showcased in Figure 2, to present a literature survey on approaches to X-IDS.

IV. APPROACHES TO EXPLAINABLE IDS (X-IDS)

As per the survey overview presented in Section III, and the taxonomy showcased in Figure 2, we will now describe in detail the black box and the white box approaches to XAI in intrusion detection systems.

A. BLACK BOX X-IDS MODELS

Guidotti et al. [65], describes a black box predictor as ''a data-mining and machine learning obscure model, whose internals are either unknown to the observer or are known but are uninterpretable by humans''. A black box model is not explainable by itself. Therefore, to make a black box model explainable, we have to adopt several techniques to extract explanations from the inner logic or the outputs of the model.

To survey the IDS landscape with respect to explainability, we have further divided the literature into different categories of XAI black box models: feature based, perturbations based, decomposition based, and hybrid approaches. These classifications are based upon how explanations are generated. A detailed literature overview is also available in Table 1.

1) FEATURE BASED APPROACHES

One popular scheme for explanations considers the influence features have on prediction. Such schemes are called feature explanations. Existing processes, such as feature engineering and feature selection, are already common in machine learning pipelines. Therefore, it is natural that features emerge as a method of explainability. Several candidate solutions that currently exploit this assumption are Partial Dependence Plot (PDP), Accumulated Local Effects (ALE), H-statistic, and SHAP.

An important generalizable SHAP-based framework is proposed by Wang et al. [11]. Their framework uses both local and global explanations to increase the explainability of the IDS model. The IDS model consists of a binary Neural Network (NN) classifier and a multi-class NN classifier. To generate explanations, both models and predictions are fed to the SHAP module. Local explanations are generated by choosing an attack and randomly selecting 100 of the occurrences. An average Shapely value is calculated, and the SHAP module outputs a confidence score for the prediction. The authors evaluate explainability by using a neptune attack, where a flooding of SYN packets is observed. The explanation results show that the top four features are related to DoS and SYN flood attacks. Using the global explanation produced by the SHAP module, researchers can make inferences about how the model might react during a related attack. However, the model's confidence seems to favor attacks that attempt many network connections (e.g. probe or DoS) over other attacks, such as privilege escalation attacks. The IDS system along with the SHAP explanations are relevant to assist subject matter experts in making security decisions.

In another effort, Islam et al. [98] built a domain knowledge infused explainable IDS framework. Their architecture is composed of two parts: a feature generalizer that uses the CIA principles and an evaluator that compares the black box models using different configurations.

The feature generalizer first maps the top three ranked features to attack types, then maps attack types to the CIA principles. For example, DoS attacks are associated with availability; Heartbleed or PortScan attacks are associated with confidentiality. Using this mapping system, the authors add three new features: C, I, and A. These three features include the aggregate scores of their related features from a data sample. If a feature positively affects a prediction, then it adds to the score; otherwise, it subtracts from the score.

The evaluator, on the other hand, runs four different feature configurations. The first configuration uses the full, preprocessed CICIDS dataset of 78 features. The second is a feature selected dataset of 50 attributes. The final two datasets are domain knowledge based: a 22 feature dataset of domain infused features and a three feature dataset consisting of C, I, and A scoring features.

Tests are run on ANN, SVM, Random Forest (RF), Extra Trees (ET), Gradient Boosting (GB), and Naive Bayes algorithms (NB). The authors outline two types of tests: explainability and generalizability. The first two datasets are used to find a baseline to compare against the authors' novel, domain infused approach. Their findings from initial experimentation show that the RF using the full dataset outperforms all other algorithms with an F1-score of 99.68%. The domain infused and CIA datasets are able to obtain an F1-score of 99.32% and 93.84% on RF and ET algorithms, respectfully. The small difference between the full dataset and the domain infused dataset show that the authors can now create a way to explain predictions without negatively impacting model performance. The authors create another CIA scoring formula that shows how much impact a CIA mapped feature had on the samples prediction. These C, I, and A scores can then be shown to an analyst to explain the prediction. To test their method against unknown attacks, the models are trained on all attacks in the dataset except one. The classifier is tested on a dataset that includes all of the attacks.

The results show that the novel domain infused dataset performs similarly to the full dataset. In one case, the domain infused dataset is able to be used to find an attack that the full dataset configuration could not. The authors have demonstrated that creating an explainable algorithm and dataset can be useful for both accuracy and explanations.

Sarhan et al. proposes another feature based technique [99]. Two feature sets, NetFlow and CICFlowMeter, are evaluated across three datasets. When new IDS datasets are created, they are not necessarily created using the same tools. NetFlow and CICFlowMeter based IDS datasets collect different feature sets. The authors test these different feature sets using Random Forests and Deep Feed Forward algorithms. The results from this experiment show a minor improvement from the NetFlow feature set over the CICFlowMeter set. The most interesting result is the change in false positives between the two feature sets. NetFlow offers a much lower false positive rate than its counterpart in many of the tests. Additionally, NetFlow is slightly faster to make predictions than CICFlowMeter. The authors conclude that NetFlow offers slightly higher quality security features. Explainability is achieved in the form of SHAP. SHAP is used to determine which features are causing this difference in performance. The authors conclude that there are certain features across all datasets that contain more security focused data. However, the most important features vary across datasets. This is attributed to the fact that each dataset has different attacks. The authors work shows the importance of feature selection during dataset creation.

A novel method in [100] uses Auto-Encoders (AE) in combination with SHAP to explain anomalies. Anomalies

TABLE 1. An overview of the existing literature on black-box approaches to intrusion detection systems, with a focus on their scope, contribution, and limitations.

Paper Title	Focus/Objective	Contribution	Limitation
Feature based IDS			
An Explainable Machine Learning Framework for Intrusion Detection Systems [11]	Locally and globally explainable NN using SHAP for IDS.	• Framework that creates both local and global explanations. • First use of SHAP in the field of IDS. • Comparison between one-vs-all classifier and multi-class classifier.	• More intrusion detection datasets should be tested. • SHAP cannot work in real-time. • SHAP needs to be tested on more robust attacks.
Domain Knowledge Aided Explainable Artificial Intelligence for Intrusion Detection and Response [98]	Use CIA principles on data to improve both generalizability and explainability of a model	• Method for the collection and use of Domain Knowledge in IDS • Use CIA principles to aid in explainability • Domain Knowledge increase generalizability	• Domain Knowledge is applied to a specific dataset. New mappings may be needed on new datasets. • More datasets need to be tested.
An Explainable Machine Learning-based Network Intrusion Detection System for Enabling Generalisability in Securing IoT Networks [99]	Explore explainability in IDS by comparing two different IDS feat.	• Evaluate two different Network Intrusion Detection datasets: NetFLow and CICFlowMeter. • Creation of two new datasets in the CICFlowMeter format. • An explainable analysis is performed using SHAP.	• Explanations are only done using SHAP. • No analysis on the performance of the explainer.
Explaining Anomalies Detected by Autoencoders Using SHAP [100]	Use SHAP to create custom explanations for anomalies found with an autoencoder.	• Method for explaining anomalies found by an autoencoder. • Preliminary experiment with real word data and domain experts. • Suggest methods for evaluating explanations.	• Custom explanation lacks any form of visualization to aid the user.
Perturbation based IDS			
A New Explainable Deep Learning Framework for Cyber Threat Discovery in Industrial IoT Networks [101]	Explainable Intrusion Detection in the field of IoT	• Conv-LSTM-based autoencoder for time-series attacks • Detects zero-day attacks • Sliding window technique that increases accuracy of CNN and LSTM model XAI concepts to improve trust	• Tested only on a single dataset • Considers only univariate time-series data
An Adversarial Approach for Explainable AI in Intrusion Detection Systems [6]	Explain models and predictions through an adversarial approach	• Methodology explaining incorrectly classified samples to help improve flaws in the model	• Only tested on DoS attacks from NSL-KDD
Feature-Oriented Design of Visual Analytics System for Interpretable Deep Learning Based Intrusion Detection [102]	A suite of visual tools used to improve explainability of CNNs	• Analysis of Features and Requirements to improve visual analysis of XAI • IDSBoard, a GUI for understanding Deep Learning Intrusion Detection • Demonstrate the effectiveness of visual analytics	• Only tested on a single dataset • Scalability of visual analytics system • Visual analytics system only designed for CNN
Explanation framework for Intrusion Detection [103]	Explaining IDS explanations using a Counterfactual technique.	• Explaining classifications based on feature importance. • Advice on how to change a classification to its desired result. • Outline the decision process so that the user can simulate it themselves.	• Analysis of the counterfactual technique was only run on one type of ML algorithm.
Decomposition/Gradient based IDS			
Toward Explainable Deep Neural Network Based Anomaly Detection. [104]	Initial steps into XAI for DNN Intrusion Detection	• Framework for creating an explainable Deep Network XAI concepts to improve trust	• Experiments are run using only DOS attacks from the NSL-KDD dataset

TABLE 1. ***(Continued.)*** **An overview of the existing literature on black-box approaches to intrusion detection systems, with a focus on their scope, contribution, and limitations.**

Towards explaining anomalies: A deep Taylor decomposition of one-class models. [105]	Explaining anomalies found by a SVM using Deep Taylor Decomposition.	• A method for 'neuralizing' a one-class SVM to be explained by Deep Taylor Decomposition.	• Experiments solely run using one-class SVM. No comparison to 'real' neural networks.
Hybrid IDS			
Achieving explainability of intrusion detection system by hybrid oracle-explainer approach [106]	Building a hybrid IDS based around 'XAI Desiderata' that does not decrease performance or add vulnerability.	• An explainer module modeled after the 'XAI Desiderata.' • A Hybrid-Oracle explainer Intrusion Detection System.	• Two models need to be effectively trained. • Would benefit from being tested on multiple datasets.
Explainable deep few-shot anomaly detection with deviation networks [107]	An anomaly detection system able of detecting anomalies learned from few anomalous training samples.	• Prior-driven anomaly detection framework. • DevNet, an anomaly detection framework based on Gausian prior, Z-Score-based deviation loss, and multiple instance learning. • A theoretical and empirical analysis of Few-shot anomaly detection.	• Experiments only run using image based datasets with relatively small sample sizes.

are detected using the reconstruction score of the AE. Samples that return a higher reconstruction score are considered anomalous. An explainer module is created with the goal to link the input value of anomalies to their high reconstruction score. Features are split into two sets. The first set contains features that are causing the reconstruction score to be higher, while the second set does the opposite. The authors label these sets 'contributing' and 'offsetting', respectively. Contributing features will have a SHAP score that is negative, and the opposite is true for offsetting. Explanations are presented in the form of a color-coded table where darker values are more important than lighter values. This novel approach to explaining AE can be improved with more iterations of its visualization style and methodology.

In another piece of work, Dang, Q. V. [108] suggests an explainable IDS that uses the eXtreme Gradient Boosting (xgboost) classifier as its base predictor model and makes explanations using PDP plots and the SHAP value. The author uses the CICIDS2017 dataset to train and test the proposed model. The experiment result indicates the proposed classifier has high detection accuracy. However, it requires high computational power. To reduce the computational need, the author utilizes the PDP plots to recursively remove features that cannot be explained without affecting their predictive accuracy.

2) PERTURBATION BASED APPROACHES

Perturbation based approaches make minor modifications to input data to observe changes in output predictions. Their explanations are based on the inclusion, removal, or modification of a feature in a dataset. These approaches are model agnostic (see Section II), therefore, they can be applied to any model.

A representative work by Wu et al. [102] showcases the advantages of this approach. The authors have created a CNN model along with a dashboard user interface (UI) to make the black box deep learning components more explainable. They gather feature requirements for their dashboard from literature. These include: (i) it is important to know the role that individual neurons play in predictions; (ii) multiple models should be tested, and the best parameters should be selected to achieve the best accuracy; (iii) visualization should assist in finding interesting results; (iv) there should be an explanation as to how the model made a decision; (v) we should be able to see the data representation in each layer of the model.

The authors use the NSL-KDD dataset to test their CNN. NSL-KDD is encoded into a 12 × 12 grayscale image that serves as input. Their model is able to achieve an 80% accuracy. The dashboard UI is able to showcase a variety of visualizations that assists in explainability. The UI includes: a detailed view of each cluster of neurons and the associated feature class, a t-SNE scatterplot of the activation values, a feature map of the convolutional kernel, a feature panel that explains how the model came to a prediction (utilizing LIME and a Saliency chart), a confusion matrix of predicted instances, and a graph for finding input data patterns. The authors demonstrate the advantages of using the dashboard UI by comparing CNNs with fewer layers than their proposed architecture. For example, the last layer in a smaller CNN shows that it is unable to detect one of the attack types (u2r) from the NSL-KDD dataset, while the proposed architecture can detect the attack. The dashboard UI is able to demonstrate that the smaller model may need more layers to be effective.

Khan et al. [101] propose an explainable autoencoder-based detection framework using convolutional and recurrent networks to discover cyber threats in IoT networks. The model is capable of detecting both known and zero-day attacks. It leverages a 2-step, sliding window technique that is used to transform a 1-dimensional (1D) sample into smaller contiguous 2-dimensional (2D) samples. This 2D sample is then fed through a CNN, comprised of a 1D convolutional layer and a 1D max-pooling layer which extracts spatial features. The data is then fed into the auto-encoder based LSTM that extracts temporal features. Finally, the DNN uses the extracted representation to make predictions. To make the model explainable, the authors use LIME [71] (see Section II-A). The dataset used for experimentation was from a real-world gas pipeline system. It consists of system logs that include packet data used to communicate with the pipeline, along with features such as packet length, pressure setpoint, and PID gain. The authors obtain a 99.35% accuracy using their proposed model. LIME shows that there are five features in the dataset that are primarily responsible for the different predictions.

In another impactful work [6], the authors argue that rather than explaining every prediction, it is possible to create a model that explains misclassifications using a *counterfactual technique*. The goal is to explain adversarial attacks, which aim to confuse models into misclassifying input samples. Using this technique, the authors find weak points in their model and develop strategies to overcome these limitations. When an input sample is classified incorrectly, minimal changes are made to the sample until it is classified correctly. The difference between the original, incorrectly labeled sample and the new, correctly labeled sample are used to explain the occurrence of the misclassification.

NSL-KDD dataset is used to create these models. A linear classifier and a multi-layer perceptron (MLP) are used during testing and the authors achieve an accuracy of 93% and 95%, respectfully. t-SNE is used to visualize the misclassified and corrected samples. The authors technique for minimizing the difference between samples is effective as the projections created by t-SNE are nearly identical. More insight can then be gathered from these projections as they show which features caused the misclassification along with the magnitude of the impact. This method appears to be a good way to communicate why a classification occurred and allows for a user to make the necessary inferences.

Burkart et al. [103] proposes a similar application of counterfactuals on an explainable IDS framework. Here the goal of the system is to answer the question: *Why did X happen and not Y?* The authors aim to create explanations that are *understandable* and *actionable*. By understandable, they mean explaining an instance of classification, and by actionable they mean giving advice for changing the classification. The framework should also allow the users to simulate these changes themselves. The counterfactual technique is used to achieve these goals.

The technique takes a vector x and locates a similar co-ordinate position $x\prime$ in the feature space, that causes a change in the predicted label. $x\prime$ should be a sample that is very similar to x. The authors' method for their explainable technique has 5 phases. In Phase 1, their algorithm finds the first counterfactual point by using an optimization problem. Phase 2 extrapolates that point by finding other points near it that are also opposite of the original vector x. By adding more than one counterfactual point, the algorithm can help find a better general understanding of the feature area. In their approach, the authors use *MagneticSampling* to achieve this goal. This set of points is used in Phase 3 to find the decision boundary. Phase 4 takes this approximated decision boundary and trains a *surrogate explainer model* for samples on both sides of the decision boundary. Phase 5 is the culmination of all of the previous work resulting in explanations, which include a feature importance explanation, a relative difference explanation, and a surrogate visualization module. The surrogate visualization can be done in a variety of ways; however, the authors choose to use a white box decision tree. The fidelity of their explainer is tested against LIME. Additionally, their method tests 2 varieties of explainers: a decision tree explainer and a linear explainer similar to LIME. The authors method performs better than LIME when *MagneticSampling* is used in Phase 2, but performs worse than LIME when random sampling is used. The tree performs better than the linear method and the authors believe that it is superior based on its performance and inbuilt explainability.

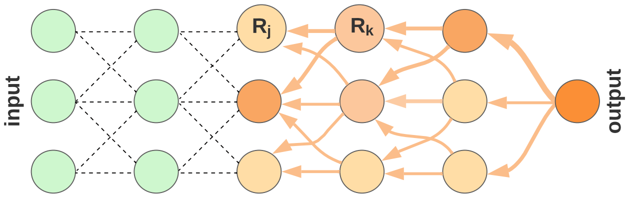

FIGURE 3. A visual depiction of Layer-wise Relevance Propagation. Relevance scores (R_j, R_k) are calculated backwards from the output for each layer (j and k represent neurons). Scores from each previous layer are used to score the next set of neurons with the final outcome being the importance of each input [110].

3) DECOMPOSITION BASED APPROACHES

Decomposition based approaches decompose the output of a model to create a relevance score. Layer-wise Relevance Propagation (LRP) is a technique where the scoring mechanism propagates backwards from the output node, highlighting activated neurons that impact predictions. According to the authors in [109], these approaches can either decompose the output or decompose the gradient of the model.

An explainable DNN using LRP has been proposed in [104]. The goal of this system is to give a confidence score of a prediction, give a textual explanation of a prediction, and the reasons why the prediction was chosen. Online, a user can see that an anomaly has been found and why it is considered an anomaly, while offline an expert can evaluate the explanations. The authors argue that the explanation for detected anomalies is provided to reduce the 'opaqueness' of DNN model and enhance 'human trust' in the algorithm. For their experiment, the authors create a partial implementation of their framework consisting of a Feed Forward DNN with explanations created by LRP. NSL-KDD is used for their experimentation. The tests are run using 4 different DNN configuration: two with three hidden layers and two with four hidden layers. Additionally, the dataset was separated into a 'simple' dataset (a smaller number of features) and a 'complete' dataset (all the features). The authors were able to achieve up to 97% accuracy from each of the implementations. The model performed better with the complete dataset rather than with the simple dataset. The authors argue that the explainability of the simple dataset is worse than the complete dataset. This is because LRP chose a feature that would be difficult for a domain expert to verify. For example, binary features like 'flag' are more difficult to explain than continuous features. The most important features for the complete dataset contained continuous values that could more easily be determined to be anomalous (src byte count and destination host count). Although the authors do not create a complete implementation with full textual explanations, their methodology could prove useful to improving the trust of regular users.

To address the issue of decomposability of DL models, the authors in [111] propose an IDS system, based on CNNs, called GRACE (GRad-CAM-enhAnced Convolution neural nEtwork). They generate visual explanations for CNN decisions by utilizing the Gradient-weighted Class Activation Mapping (Grad-CAM) [96].

The authors use three different datasets including KDD-CUP-99, NSL-KDDCUP99, and UNSW-NB15 to train their model. The textual dataset is transformed using image encoding which converts the training sample from the 1D feature vector form $\mathbf{X}^{1D}$ with size 1 $\times M$ to the 2D image form $\mathbf{X}^{2D}$ with size m $\times m$ (with $M \leq m_2$) and fed to the CNN model. The final convolution layer of 2D CNN is used to create heatmaps of class activations on input images, i.e., 2D grids of scores. Each pixel in the grid represents traffic characteristics, e.g., Destination port (X1) or idle max(X77). The scoring mechanism demonstrates how important each pixel (feature) is to a specific output class. This understanding of the most important features aids in the feature engineering process, resulting in a CNN model with higher accuracy.

To evaluate the performance of the model three evaluation metrics are used such as F1-Score (F1), Accuracy (A), and Computational complexity (T) (time spent to train the model). Of these, F1 and A metrics are used to compare against state-of-the-art such as CNN, GAN, LSTM, RNN, Triplet, DNN, MLP, and Autoencoder. Experimental results suggest GRACE generally outperforms its competitors. However, there are a few exceptions where the proposed model suffers slightly. For instance, when using the NSL-KDD dataset, the Triplet methods obtain 86.6% and 87.0% of A and F1, respectively, compared against 85.7% and 86.8% of the proposed model. The authors argue that this explanation approach can aid in the development of a more robust intrusion detection model.

Kauffmann et al. [112] propose another decomposition strategy aimed at verifying that a 'Clever Hans' strategy has not been adopted by the ML model. LRP is leveraged as an explainer module to aid in discovering this phenomenon. Three separate models are trained: a kernel density estimator, an autoencoder, and a deep one-class model. Image based anomaly detection datasets MNIST-C and MVTec are used for this experiment. A 'Clever Hans' score is adopted that is simply the difference between the detection accuracy and explanation accuracy. Detection accuracy is the ROC score, and explanation accuracy is the cosine similarity between the ground-truth and the pixel-wise explanation. It renders a score between 1 and -1 where 1 expresses a 'Clever Hans' phenomenon. Results from their testing show that, based on their scoring system, all of the models show some form of 'Clever Hans' logic. To address this problem, the authors propose a method of bagging anomaly detectors. This solution does not remove the phenomenon, but it does help to reduce it.

The previous authors also explore Deep Taylor Decomposition (DTD) for model explainability [105]. DTD is a technique that decomposes each neuron in each layer to determine feature relevance. A 'neuralized', one-class SVM is proposed that can be explained using DTD. The 'neuralized' form is a mapping of distance between the original sample and the SVM created support vectors as the first layer. The second layer is a soft min-pooling layer that calculates the 'outlierness' of samples. Samples can then be explained using DTD by decomposing each of the neurons in the prediction. In their experiment, they use image based datasets for finding anomalies. DTD is used to highlight anomalous pixels in each image.

4) HYBRID APPROACHES

A hybrid black box predictor, white box explainer has been created by Szczepanski et al. [106]. Their framework is built with principles from the "XAI Desiderata": Fidelity, Understandability, Sufficiency, Low Construction Overhead, and Efficiency [113]. The authors aim to contribute a system that is reliable, easy to understand, flexible, and meets all previous criteria without losing accuracy. With these goals in mind, a framework that uses local explanations is created. Their framework includes an ANN that predicts samples and a white box explainer that takes the output of the ANN and the original sample as input. The explainer is model agnostic and replaceable with any other explanation algorithm. The authors' explainer uses a clustering algorithm that uses a heuristic called Mean Distance to Average Vector. Clustering is done based on all of the attributes except the label. n centroids are computed for all features, then a model is trained for each centroid cluster created. Another distance based algorithm is used to find a centroid cluster that is both close to the predicted sample and gives the same prediction as the ANN. The selected cluster is then used as a visualized explanation for a prediction. The authors note that it is possible that the explainer may not return a valid tree and that the model should be trained on a feature rich, diverse dataset. The authors experiment using the CICIDS2017 dataset. The ANN is able to achieve an accuracy of 98%, and the explainer is able to achieve an accuracy of 99% with 200 clusters. The authors have created a system where there are effectively two predictors that are used to confirm and explain the other's prediction.

Pang et al. [107] create a framework based on Few-Shot Anomaly Detection (FSAD). The authors claim that their framework is interpretable and explainable through a probability based scoring method and an image demonstrating anomalous areas found in samples. One of the problems faced in IDS/Anomaly Detection is that models are generally trained on unsupervised, normal data. This makes it difficult for models to discern from normal and anomalous data. The authors aim using FSAD to improve detection rates. However, FSAD has difficulties learning a generalized representation of anomalies from a few samples and it is challenging to learn a robust representation of data with respect to anomolous data. To resolve this, the framework needs to be able to learn about anomalous samples but not learn that all anomalies are the same as the training samples. The authors achieve this by using a prior driven anomaly score and end-to-end optimization of anomaly scores with deviation learning based on the prior probability. The architecture of DevNet is composed of an Anomaly Scoring Network and a Reference Score Generator that outputs into a Multiple-Instance-Learning-based (MIL) deviation loss Score Learner. The Anomaly Scoring Network is a function ϕ that creates a scalar anomaly score for pieces of an input. In this case, the pieces of an input are parts of an image. The Reference Score Generator creates a reference score μ_r, which is a mean score of randomly selected non-anomalous samples. The reference score is derived from a prior F. The function $\phi(X)$, μ_r, and the standard deviation of μ_r are provided as input into the MIL Deviation Loss Learner whereby the goal is to optimize anomaly scores so that anomalies deviate significantly from normal samples.

The framework is tested on a variety of image datasets for identifying defects, planetary bodies, and medical anomalies. DevNet is tested against five other models and performs better on 7 out of 9 datasets. DevNet is able to achieve an AUC score between 80% to 98% amongst all of the datasets. As for explainability, the authors demonstrate that the algorithm can display the anomalous region on an image. DevNet generates both a black-white image of the location of the defect and an overlaid image showing where the defect lies on the original image.

B. WHITE BOX X-IDS MODELS

Models that can provide an explanation to expert users without utilizing additional models are referred to as *interpretable* or *white box models* [88]. A white box model's internal logic and programming steps are completely transparent, resulting in an interpretable decision process [114]. However, when the model is to be explained to non-expert users, it may demand post-hoc explainability, such as visualizations [64]. This interpretability, on the other hand, usually comes at a price in terms of performance [115].

A myriad of white box approaches are available for intrusion detection. Our survey will focus on the approaches most commonly used in the literature, as per our overview presented in Section III and the taxonomy showcased in Figure 2. Table 2 summarizes state-of-the-art research, challenges, and contributions with respect to white box approaches for intrusion detection systems.

1) REGRESSION

Linear Regression (LR) is a supervised ML technique that establishes a relationship between a dependent variable and independent variables by computing a *best fit* line. The linearity of the learned relationship puts LR under the umbrella of interpretable models.

Various regression-based IDS models exist in the literature. Subba et al. [122] deployed anomaly-based intrusion detection systems using two different statistical methods: Linear Discriminant Analysis (LDA) and LoR. While LR models are desirable for intrusion detection purposes, their performance is susceptible to outliers [123]. To mitigate the impact of outliers, the same authors proposed a robust regression method for anomaly detection [124]. The proposed method uses heteroscedasticity and a huber loss function instead of homoscedasticity and sum of squared errors.

While the existing approaches render promising outcomes, none of them were designed with *explainability* in mind. To overcome the issue of *explainability* in the area of *hardware performance counter* (HPC) – based intrusion detection, Kuruvila et al. [116] propose an explainable HPC-based Double Regression (HPCDR) ML framework. The study examines two distinct types of attacks: microarchitectural and malware. For the first type of attack, tests are conducted on five distinct datasets: Rowhammer, Flush+Flush, Spectre, Meltdown, and ZombieLoad. For the second attack, two distinct datasets are considered: Bashlite and PNScan. To minimize computational overhead, the proposed study employs Ridge Regression (RR) rather than Shapely values to generate interpretable results. First, the three ML models (RF, DT, and NN) are chosen to evaluate the classification accuracy. Second, the output from these models is perturbed and passed to the first RR model where HPCs are employed as features and weight coefficients are received. These furnished coefficients are run on the second RR model, which identifies the most malicious sample. The authors argue that by utilizing double regression techniques, their proposed method provides transparency, which enables users to locate malicious instructions within the program.

2) DECISION TREE AND RULE BASED

A Decision Tree (DT) is a tree structure with decision support system elements based on graph theory. In contrast to LR method, it works even when the relationship between input and output is nonlinear. In their simplest form, DT possesses three properties that make them interpretable [29]: simulatability, decomposability, and algorithmic transparency.

A simple rule is typically represented as a logical implication of IF-THEN statements by combining relational statements to form their knowledge [125]. These rules can be extracted from DT. Rule-based models are considered transparent because they generate rules to explain their predictions.

Mahbooba et al. [117] approach the task of developing an interpretable model to identify malicious nodes for IDS using a DT on the KDD dataset. They chose the Iterative

TABLE 2. A summary of the existing literature on white-box approaches to intrusion detection systems, with an emphasis on their scope, contribution, and limitations.

Paper Title	Focus/Objective	Contribution	Limitation
Regression based IDS			
Explainable Machine Learning for Intrusion Detection via Hardware Performance Counters [116]	To develop an explainable X-IDS technique based on the double RR technique and utilizing HPC as a feature.	• Proposes an explainable HPC-based Double Regression (HPCDR) framework for intrusion detection with human-interpretable results. • HPCDR is evaluated against real-world malware to determine whether it provides transparent hardware-assisted malware detection and to detect microarchitectural attacks with an indication of the malicious origin.	• DL models were not chosen to evaluate the optimal ML model. • Only Four HPC features were chosen for experimentation. • Other microarchitectural attacks (e.g. Prime+Probe) and malware (e.g. Rootkits) are not considered in the study.
Decision Tree and Rule based IDS			
XAI to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model [117]	Focused on the interpretability in a widely used benchmark dataset KDD datasets.	• Addressed XAI concept to enhance trust management that human expert can understand. • Analyzed the importance of feature based on the entropy measure for intrusion detection. • Interpreted the rules extracted from the DT approach for intrusion classification.	• Information gain in decision trees is biased in favor of those attributes with more levels. • This behavior might impact prediction performance.
A Hybrid Approach for an Interpretable and Explainable Intrusion Detection System [118]	To design interpretable and explainable hybrid intrusion detection system to achieve better and more long-lasting security.	• Providing an IDS that stands out for its ML support on populating the knowledge base. • Focus on interpretability and explainability, since it justifies the suggested rules, and the diagnosis performed to each asset.	• DT only considered as ML model for system design. • Knowledge base is small.
Statistical and Probabilistic Models			
A New Method for Flow-Based Network Intrusion Detection Using the Inverse Potts Model [119]	To develop a new method for flow-based network intrusion detection using inverse statistical method.	• Implementation of a naturally interpretable flow classifier based on the inverse Potts model to be employed in NIDS. • Performance comparison with other ML based models using three datasets.	• Only binary classification is considered in the approach. • Applicability of the proposed methods in real world data.
Clustering based IDS			
Explainable unsupervised machine learning for cyber-physical systems [120]	Propose a novel Explainable Unsupervised Machine Learning (XUnML) approach using the Self Organizing Map (SOM) algorithm.	• Brief overview of Supervised Machine Learning (SML), Unsupervised Machine Learning (UnML), and XAI. • Exploring initial desiderata towards Explainable UnML (XUnML), defining XUnML terminology based on the terminology used for XAI, and exploring the necessity of XUnML for CPSs.	• Only clustering method is used.
ANNaBell Island: A 3D Color Hexagonal SOM for Visual Intrusion Detection [121]	Provide explanation to the outputs of SOM models using color scheme and island landscape analogy for different network traffics.	• Benign and malicious traffic is separated by color coding and zoning in the island. • Color and zone categorization of network traffic provides the explanation of the output.	• It is not clear if the temporal map maintain same basic landscape or change over time. • The proposed map seems to be specific to the tested network only.

Dichotomiser 3 (ID3) algorithm to ensure interpretability because it mimics a human-based decision strategy. The authors demonstrate that the algorithm can rank the relevance of features, provide explainable rules, and reach a level of accuracy comparable to state-of-the-art. Another explainable decision tree model is proposed in [125] and [126], with the latter being an extension of work in [127].

Sinclair et al. [128] extract rules using a DT and a Genetic Algorithm (GA) for improving the performance of the IDS model. The authors in [129] and [130] focus on optimizing the IDS model by extracting rules using a GA. To add transparency to the decision process, Dias et al. [118] proposed an interpretable and explainable hybrid intrusion detection system. The proposed system integrates expert-written rules and dynamic knowledge generated by a DT algorithm. The authors suggest that the model can achieve explainability through the justifications of each diagnosis. Justification of certain predictions is provided in a tree-like format in the form of a suggested rule that provides a more intuitive and straightforward understanding of the diagnosis.

Snort is the world's most widely used open-source rule-based intrusion prevention system (IPS) [131]. It employs a set of rules that help define malicious network activity. These rules are then used to identify packets and generate alerts for users [131], [132].

3) STATISTICAL AND PROBABILISTIC METHODS

In statistics, the mean, standard deviation, and any other type of correlation are referred to as moments [133]. Statistical and probabilistic methods use this information to determine whether the given event is anomalous or not. The moment is predicted anomalous if they are either above or below a predefined interval. This approach is further divided into the univariate, multivariate, time series, parametric, non-parametric, operational and Markov models [133], [134], [135], [136].

Various IDS based on statistical and probabilistic models have been proposed. IDS based on the mean and standard deviation is explained in [137], while a study relating to multivariate modeling is proposed in [138]. Gyanchandani et al. [139] proposed an IDS based on the Markov process.

A different approach to intrinsically explainable statistical methods for network intrusion detection is proposed by Pontes et al. [119]. They introduce a novel Energy-based Flow Classifier (EFC) that utilizes inverse Potts models to infer anomaly scores based on labeled benign examples. This method is capable of accurately performing binary flow classification on DDoS attacks. They perform experiments on three different datasets: CIDDS-001, CIC-IDS2017, and CICDDoS19. Results indicate that the proposed model is more adaptable to different data distributions than classical ML-based classifiers. Additionally, they argue that their model is naturally interpretable and that individual parameter values can be analyzed in detail.

4) CLUSTERING

Clustering is the most widely used strategy for unsupervised ML. It classifies samples according to a similarity criterion. Clustering algorithms that can be explained have several advantages. The primary benefit of explainable clustering is that it summarizes the input behavior patterns within clusters, enabling users to comprehend the clusters' underlying commonalities [120]. As stated in Section III-A there are various clustering algorithms available. However, in the context of X-IDS, we will only focus on Self-Organizing Maps (SOMs).

SOMs are an unsupervised clustering technique within the artificial neural networks umbrella. It has two layers: an input layer that accepts high dimensional space and an output layer that generates a non-linear mapping of high-dimensional space into reduced dimensions. It is trained to produce a low dimensional representation of a large training dataset while preserving important topological and metric relationships of the input data [140].

An anomaly detection system using SOM techniques based on offline audit trail data is proposed in [141]. The major shortcoming of the proposed system is it does not allow for real-time detection. On the other hand, the authors in [142] propose Hierarchical SOMs (HSOM) for host-based intrusion detection on computer networks that are capable of operating on real-time data without requiring extensive offline training or expert knowledge. Another model based on HSOM is proposed in [143]. Wickramasinghe et al. [120] developed a novel model-specific explainable technique for the SOM algorithm that generates both local and global explanations for Cyber-Physical Systems (CPS) security. They used the SOMs training approach (winner-take-all algorithm) together with visual data mining capabilities (Histograms, t-SNE, Heat Maps, and U-Matrix) of SOMs to make the algorithm explainable.

A 3D color hexagonal SOM for visual intrusion detection called ANNaBell Island is proposed in [121] which is an extension of 1D ANNaBell reported in [144] and [145]. To make the SOM process and its output explainable to the users, the authors designed a hexagonal SOM in a meta-hexagonal layout, referred to as an island, that graphically displayed features of network traffic. The output of the SOM model was used to create a color-separated 3D landscaped island that represents various types of network traffic, distinguishing between malicious and normal behavior.

After surveying the current black box and white box approaches to X-IDS, we propose in the next section, a generic explainable architecture with a user-centric approach for designing X-IDS that can accommodate a wide variety of scenarios and applications without adhering to a specification or technological solution.

V. DESIGNING AN EXPLAINABLE IDS (X-IDS)

The purpose of an IDS is to continuously monitor a network for malicious activity or security violations known as incidents of intrusion. If found, intrusions are reported to the cybersecurity professional responsible for monitoring such systems. A significant problem with AI based IDS is their high false positive and false negative rates. Recently, many IDS based on ML/DL techniques have been proposed to address this issue, such as DNN [33], [146], RNN [147], [148], and CNN [149], [150]. These techniques yield unprecedented detection accuracy. However, the effective use of these approaches require using high-quality data, as well as a considerable amount of computing resources [151]. Additionally, this modeling approach has

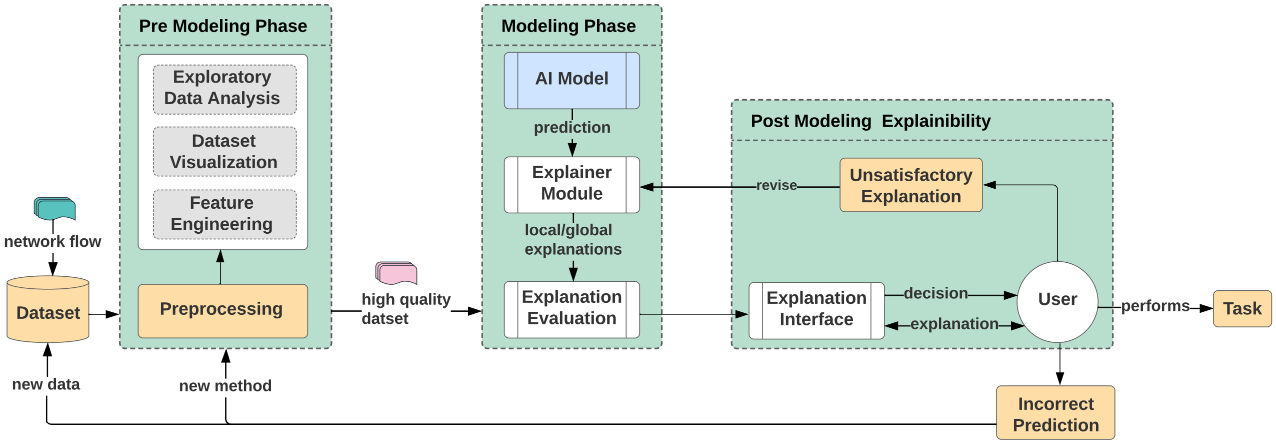

FIGURE 4. Recommended architecture for the design of an X-IDS based on DARPA [24]. The layered architecture is divided into three phases: pre-modeling, modeling, and post-modeling explainability. Each phase contributes to the development of an explanation for various stakeholders, thereby assisting in decision-making.

typically suffered from model bias, a lack of decision process transparency, and a lack of user trust.

The IDS systems based on ML/DL techniques are designed to generate event logs in the form of 'benign' or 'malicious' classification reports, that can be further analyzed by CSoC analysts. However, they do not showcase the connection between the inputs and output (i.e., they fail to indicate the reasoning behind the decision). To be more precise, a cybersecurity specialist serves as a user who reviews IDS results, but is not a component of the intrusion detection process [152]. In turn, this creates a larger problem for CSoC experts, as they are unable to optimize their decisions based on the model's decision process.

To address this semantic gap, one promising technique is to design X-IDS with a human-in-the-loop approach. Typically, methods that are retraceable, explainable, and supported by visualizations amplify cybersecurity analysts' understanding in managing cybersecurity incidents in both proactive and reactive manners.

In the following sub-sections, we explain the recommended architecture, as depicted in Figure 4, that could be used as guidance to design an Explainable Intrusion Detection Systems (X-IDS). The X-IDS architecture proposed in this paper is based on the DARPA recommended architecture for the design of XAI systems [24]. The layered architecture consists of three phases: *pre-modeling phase*, *modeling phase*, and *post-modeling explainability phase*. In each phase, different modules work in tandem to provide CSoC analysts with more accurate and explainable output. We believe that this architecture is sufficiently generic to accommodate a variety of scenarios and applications without adhering to a particular specification or technological solution.

A. PRE-MODELING PHASE

The first phase is a pre-modeling phase. The input of this module is raw network flow (dataset) and the output is a high-quality dataset. In the following subsections, we will first describe different benchmark datasets available for Intrusion Detection. We then present common data preprocessing techniques used in the literature.

1) DATASETS

While access to representative, labelled datasets for cybersecurity related AI tasks remains a challenge, a variety of publicly accessible datasets can be used to train and benchmark X-IDS. These are unprocessed network flows extracted from packet captures. To address privacy concerns, many of these datasets are generated in an emulated environment. NSL-KDD [153], based on the KDD-CUP-99 [154], is a dataset frequently present in the literature. Although old, its use allows comparisons with previous works. NSL-KDD is relatively small compared to other datasets in the field. A more modern dataset is CICIDS2017 [155], which contains more up-to-date attacks and network flows. In addition, it includes 3 million samples, which allows scalability testing. Another noteworthy dataset is UGR [156], a multi-terabyte dataset collected over the course of 5 months. This dataset is built to test IDS for long-term trends. The authors state that their dataset captures potential trends in daytime, nighttime, weekday, and weekend traffic.

These publicly available datasets, though good for benchmarking, are not suitable for deployable systems. We recommend, CSoC users deploying X-IDS systems evaluate these systems on organizational representative datasets.

2) EXPLORATORY DATA ANALYSIS (EDA) AND DATA VISUALIZATION

Data preprocessing is essential for increasing the likelihood of ML models producing accurate predictions. Using Exploratory Data Analysis (EDA), one can gain a general understanding of a dataset's key features and characteristics. To comprehend the features, visualization techniques such as heat maps, network diagrams, bar charts, and correlation matrices may be used. Once a comprehension of feature space has been attained, the data is forwarded to the feature engineering model for further processing.

3) FEATURE ENGINEERING

The general trend in preprocessing IDS datasets is to normalize the numerical features and to One-Hot Encode (OHE) the categorical features. After the datasets are encoded, their feature space can be quite large which makes them computationally expensive. Two approaches to reducing dimensions are widely discussed in the literature: Feature Selection and Feature Extraction.

Feature selection techniques are used to reduce the feature space by selecting a subset of features without transforming them. There are three types of feature selection techniques popular in the IDS domain: filters, wrappers, and the embedded/hybrid method [157]. Apart from these, libraries such as Scikit-Learn [158] have also been used in published works for feature selection.

Another technique used in feature engineering is feature extraction (also known as dimensionality reduction). Feature extraction reduces the size of the feature space by transforming the original features while retaining most of their defining attributes. The most commonly used feature extraction technique in the literature is the Principal Component Analysis (PCA) [159]. PCA is an unsupervised method that does not require class knowledge to identify features. It also facilitates the identification of correlations and relationships between the features of a dataset.

B. MODELING PHASE

The second phase is the modeling phase. The input of this phase is the high-quality dataset generated in the pre-processing phase and the output is the explanations generated by the explainer module. First, the high-quality dataset is fed into the ML/DL model of choice. Second, the predictions generated by the model in use are passed through an explainer module. Third, these explanations are evaluated by an evaluation module. This process enables users to understand the reason behind certain predictions, which in turn, helps the CSoC analysts in their decision-making process.

1) AI MODEL

In Section IV, we discussed two different approaches which are currently being employed by different authors to create X-IDS: the black box and the white box. AI modules in these approaches generate predictions. However, there is a trade-off between the accuracy and the interpretability with these approaches. The white box approaches are popular for their interpretability, while the black box approaches are known for their prediction accuracy. In context of IDS, high prediction accuracy is required to prevent attacks. Moreover, black box models can capture significant non-linearity and complex interactions between data that white box models are not able to capture. For example, Recurrent Neural Networks (RNN) can capture temporal dependencies between samples. On the other hand, models like Support Vector Machine (SVM) and Deep Neural Network (DNN) can create their own representation of data. which might be helpful to discover unknown attacks. For this reason, we believe that future X-IDS should be built using black box models.

In our literature review, we found that authors use a variety of black box algorithms in their work, such as SVM, CNN, RF, and MLP, which prove to be quite effective. Another popular algorithm of choice in the intrusion detection domain is a variant of the RNN, referred to as Long Short-Term Memory (LSTM). Recently, Generative Adversarial Networks (GAN) have also become relatively popular. Consequently, there are a multitude of black box algorithms from which to select. Explainer modules then approximate the prediction generated by AI module employing a white box or black box algorithms.

2) EXPLAINER MODULE AND EVALUATION

The prediction generated by the model of choice in the AI module is then fed to the explainer module. The common explainers used from previous works include LIME, SHAP, and LRP. These out-of-the-box modules allow for quick testing on different algorithms and datasets. However, there are some problems with solely using these approaches in future X-IDS works. To begin, methods such as SHAP do not run in real-time. Therefore, it may be time-consuming to attempt to use SHAP on a simple Multi-Layer Perceptron classifier with a large feature space dataset. In X-IDS, both predictions and explanations must be made as quickly as possible. Secondly, these approaches are not always designed with X-IDS stakeholders in mind.

At present, there are no set standard metrics to evaluate explanations. Several authors have attempted to evaluate explanations in various ways. In Section II-C we described different ways to evaluate the explanations. Metrics such as application grounded evaluation, human-grounded evaluations, and function-grounded evaluation proposed by Doshi et al. [81] can be used as a baseline to evaluate the explanation generated by X-IDS. A noteworthy method to evaluate the effectiveness of explanations is proposed by authors in [24]. Figure 6 illustrates their approach.

C. POST MODELING EXPLAINABILITY PHASE

The third and final phase is the post-modeling explainability phase. This phase has two major components: the explanation interface and users. The recommendation, decision, or action generated by the AI module, explained by an explainer module, and evaluated by an explanation evaluation module is rendered in a graphical user interface (explanation interface). The users, on the other hand, use this interface to make an informed decision.

1) EXPLANATION INTERFACE

The custom visual dashboards are created to help the user to understand the X-IDS. An excellent approach to building such an explanation interface is found in the work by [102] and [98]. The engineers who design X-IDS can use this approach as guidance to create their explanation interface.

Furthermore, this paper also recommends that future X-IDS developers make custom explainers built for specific stakeholders to help improve explainability. Open-source toolkits and libraries are also available that create a visual dashboard and explain the prediction. One such library is Shapash [160]. It is an overlay package to other intelligibility libraries, such as Shap and Lime, that are dedicated to the interpretability of models. Another example of the library for quickly building interactive dashboards for analyzing and explaining the predictions and workings of sci-kit-learn machine learning models is explainerdashboard [161].

2) USERS

For this paper, the stakeholders will consist of developers, defense practitioners, and investors. Section VI-B discusses the need for defining the identity of the stakeholders of an X-IDS system. The developers are tasked with creating, modifying, and maintaining the X-IDS. The defense practitioners guard the assets of the investors. Lastly, the investors make budgeting decisions for the benefit of the X-IDS system and other assets. These three audiences have distinct tasks and explainability requirements that must be addressed differently by the X-IDS. An explanation interface designed from the user's perspective can bridge this gap.

If an explanation is unclear or unhelpful, the stakeholders will need a way to voice that opinion. For example, a set of explanations is too complicated for a group of investors. Investors may ask for additional explanations that simplify or even link to a web page that can teach them more on a subject. In such a situation, the developers can revise the explainer module to fit the users request or needs. This could also include making a new explainer module or updating to a new state-of-the-art module like those in AIX360 [80]. For the same reasons, incorrect predictions and explanations need to be corrected and updated. The developers or defense practitioners will then need to introduce new data to the model. Moreover, a different method of data preprocessing may be required to augment the efficacy of the model.

To make the recommended X-IDS architecture as shown in Figure 4 a reality, researchers need to study different aspects of the three phases. To this end, in the next section we discuss various challenges inherent in designing the proposed X-IDS architecture and make research recommendations aimed at effectively mitigating these challenges for future researchers interested in developing X-IDS.

VI. RESEARCH CHALLENGES AND RECOMMENDATIONS

The sub-domain of explainable AI based Intrusion Detection Systems is still in its infancy. Researchers working on X-IDS must be made aware of the issues that hinder its development. The issues that we described in Section II such as finding the right notion of explainability, generating explanations from a stakeholder's perspective, and lack of formal standard metrics to evaluate explanations are prevalent in the X-IDS domain as well. Existing X-IDS research is primarily focused on the goal of making algorithms explainable. Explanations are not being designed around stakeholders, and researchers need to quantify useful evaluation metrics. Apart from these challenges, issues pertaining to IDS may also pose a problem for X-IDS. There are many promising avenues of exploration, in this section we detail some existing research challenges and give our recommendations.

A. DEFINING EXPLAINABILITY FOR INTRUSION DETECTION

The first problem faced by researchers designing X-IDS is the lack of consensus on the definition of explainability in IDS. The research community needs to agree on a common definition of explainability for IDS. To find common ground, we can leverage the foundational XAI definition proposed by DARPA [31]. However, an X-IDS definition needs more security domain-specific elements. The inclusion of the CIA principles may be a good start for cementing a definition that combines aspects of cybersecurity and XAI.

Questions relevant to the X-IDS that researchers need to answer include: ''What is explainability when used for intrusion detection?'', ''How do we effectively create explanations for IDS?'', and ''Who are we creating explanations for?''. Other questions such as ''How can Confidentiality, Integrity, and Availability benefit from explanations?'' and ''How do we categorize X-IDS algorithms?'' should be reassessed by X-IDS researchers as well. Current work is extremely narrow in its scope and limits its objective to explaining each sample in an IDS dataset. These works also do not consider the type of audience when building X-IDS.

B. DEFINING TASKS AND STAKEHOLDERS

The second challenge is to define the task and the stakeholders of the X-IDS ecosystem. After formalizing the definition of 'explainability' for X-IDS, we need to create explanations tailored to the stakeholders. Figure 5 demonstrates a simple user and explanation taxonomy. We consider three major stakeholders based on their roles in this taxonomy including *IDS developers*, *security analyst*, and *investors*. Each of the stakeholder categories necessitates a different degree of explanation and visualization. Developers and CSoC members are more familiar with the field and may want more complex explanations. Investors and managers, on the other hand, may be more satisfied with summarized visualizations. Each user group performs varying tasks based on the explanations. Programmers will work to debug and increase the efficacy of the AI model. CSoC members will be tasked with protecting investor assets. Indirectly, investors will need to make hiring or budgeting decisions. Take for example a corporate, network security system. The set of stakeholders consists of IDS developers, security analysts, upper-level managers, and team managers. In such a scenario, IDS developers will be creating and/or updating the corporation's IDS. These developers will want the IDS to return which features from local and global explanations are making the most impact. Additionally, as attacks change over time, these

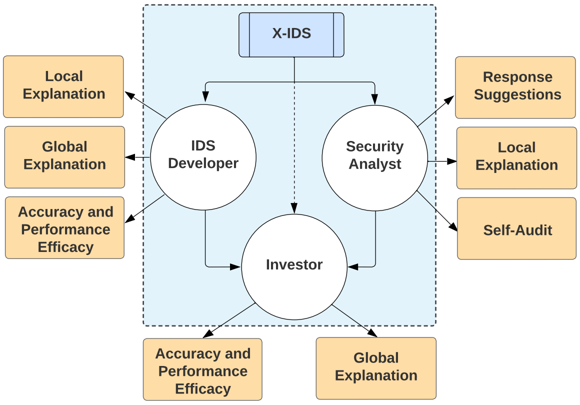

FIGURE 5. A simple taxonomy illustrating the importance of tailoring explanations to specific stakeholders based on their roles in CSoCs.

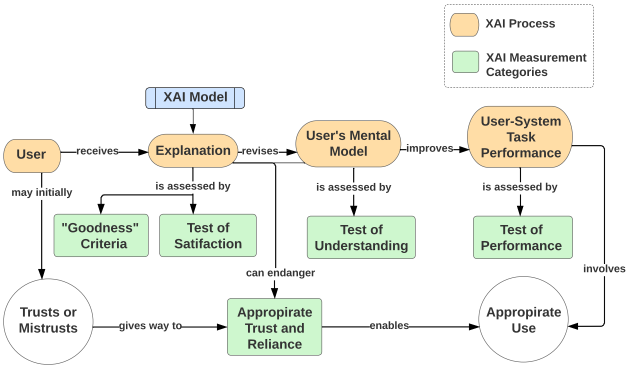

FIGURE 6. Different categories for assessing the effectiveness of explanations in the IHCM psychological model with detailed explanation process [24].

individuals will want explanations potentially keeping track of these changes. New attacks can be known to exist if the performance begins to degrade, so accuracy and performance metrics will be integral to maintaining the IDS. Security analyst or server admins would benefit in a similar way to IDS developers. Having potential leads to base their actions on could lower system down time. The different managerial levels could be using certain other metrics or explanations to aid them in leadership decisions. Hiring more staff, adding more funding, or deciding to pivot to a new system would be some actions managers could take. The needs of each group are different and future research is needed to determine the best types of explanations that will benefit each group the most.

C. EVALUATION METRICS

The third challenge in designing X-IDS is evaluating the explanation generated by the 'explainer module'. Finding the best explanation for each stakeholder category requires customized evaluation metrics. Currently, there is no consensus on metrics for explanations. In Section II-C we described a body of literature proposing various evaluation metrics that could be used towards evaluating explanations. In particular, we recommended evaluation metrics proposed by authors in [81] to evaluate explanations for X-IDS in Section V-B. Another notable work that could serve as a baseline for evaluating explanations is the psychological model of explanation created by the Florida Institute for Human and Machine Cognition (IHMC) [24]. The proposed model is illustrated in Figure 6. The user receives an explanation from the XAI model. This explanation can be tested for "goodness" and the satisfaction of the user/stakeholder. The user then revises their mental model of the XAI system. Their understanding of the system can be tested. Tasks are performed based on the explanation. The IHMC model merges the purpose of the XAI model, with the task and mindset of the user.A noteworthy method to evaluate the effectiveness of explanations is proposed by authors in [24]. Figure 6 illustrates their approach.

D. ADVERSERIAL AI

Adversarial AI refers to the use of artificial intelligence for malicious purposes, including attacks on other artificial intelligence systems to evade detection [162] or to poison data [163]. Malicious actors can potentially attack the classifiers that are used to generate predictions and cause misclassification. In context of X-IDS, the explanations generated by the explainer module may become a new point of attack for malicious actors. Attackers may add, delete, or modify explanations to evade detection [164]. Attackers may also attack training datasets to alter the explainer's behavior. The methods and effects of these attacks will need to be explored. Defense techniques must be created to correct attacked explanations. Studies to defend IDS against adversarial attack include [165], [166], and [167], etc. Study-specific to adversarial approach for X-IDS is discussed in [6].

E. MISLEADING/INCORRECT EXPLANATIONS

An explanation does not have to be attacked to be misleading. The explanation itself may be misleading, or the user may interpret the explanation incorrectly. This may lead to circumstances where the model is correct and the user is the problem. The explainer will need to be modified to prevent user error in such situations.

Explanations that are misclassified either by an attack or due to the poor quality of data can have a significant negative impact on CSoCs. CSoCs security analysts should always critically analyze the reasoning behind the prediction. Moreover, methods for auditing previously incorrect explanations should be created. Ideally, the X-IDS should be able to audit itself and generate explanations for the audit.

F. SCALABILITY AND PERFORMANCE

Performance is of utmost importance for an IDS. CSoCs can incur losses for lost time. Explanations should not needlessly slow down an IDS. So how do we optimize an X-IDS? One approach is that the explainer could generate explanations for every sample it sees, or it could strategically choose which samples to explain. A comprehensive analysis of the CPU, RAM, and disk usage should be run on current and future explainers.

VII. CONCLUSION

The exponential growth of cyber networks and the myriad applications that run on them have made CSoC, Cyber-physical systems, and critical infrastructure vulnerable to cyber-attacks. Securing these domains and their resources through the use of defense tools such as IDS, is critical to combating and resolving this issue [10], [11]. Recent AI-based IDS research has demonstrated unprecedented prediction accuracy, which is helping to lead to its widespread adoption across the industry. CSoC analysts largely rely on the results of these models to make their decision. However, in most cases, decision-making is impaired simply because these opaque models fail to justify their predicted outcomes. A solution to this problem is to embrace the concept of 'explainability' in these models. This, in turn, may facilitate quick interpretation of prediction, making it more feasible for CSoC analysts to accelerate response times.

A systematic review of current state-of-the-art research on 'XAI' or 'explainability' highlighted some key challenges in this domain, such as the lack of consensus surrounding the definition of 'explainability', the need to formalize explainability from the user's perspective, and the lack of metrics to evaluate explanations. We propose a taxonomy to address this problem with a focus on its relevance and applicability to the domain of intrusion detection.

In this paper, we present in detail two distinct approaches found in the body of literature which address the concern of 'explainability' in the IDS domain, including the white box approach and the black box approach. The white box approach makes the model in use inherently interpretable, whereas the black box approach requires post-hoc explanation techniques to make the predictions more interpretable (e.g., LIME [71], SHAP [32]).While the former approach may provide a more detailed explanation to assist CSoC members in decision-making, its prediction performance is in general outperformed by the latter. Nevertheless, the field of IDS requires a high degree of precision to prevent attacks and avoid false positives. Bearing this in mind, a black box approach is recommended when developing an X-IDS solution.

In addition, we also propose a three-layered architecture for the design of an X-IDS based on the DARPA recommended architecture [24] for the design of XAI systems. This architecture is sufficiently generic to support a wide variety of scenarios and applications without being bound by a particular specification or technological solution. Finally, we provide research recommendations to researchers that are interested in developing X-IDS.

ACKNOWLEDGMENT

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Army ERDC or the U.S. DoD.

REFERENCES

[1] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, pp. 949–961, Jan. 2019.

[2] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber–physical system security for the electric power grid," *Proc. IEEE*, vol. 100, no. 1, pp. 210–224, Jan. 2011.

[3] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. 47th Design Autom. Conf. (DAC)*, 2010, pp. 731–736.

[4] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Proc. Workshop Future Directions Cyber-Phys. Syst. Secur.*, vol. 5, 2009.

[5] M. Ahmadian and D. C. Marinescu, "Information leakage in cloud data warehouses," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 2, pp. 192–203, Apr. 2018.

[6] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable AI in intrusion detection systems," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 3237–3243.

[7] V. Cardellini, E. Casalicchio, S. Iannucci, M. Lucantonio, S. Mittal, D. Panigrahi, and A. Silvi, "An intrusion response system utilizing deep Q-networks and system partitions," 2022, *arXiv:2202.08182*.

[8] K. Sane, K. P. Joshi, and S. Mittal, "Semantically rich framework to automate cyber insurance services," *IEEE Trans. Services Comput.*, early access, Sep. 16, 2021, doi: 10.1109/TSC.2021.3113272.

[9] A. McDole, M. Gupta, M. Abdelsalam, S. Mittal, and M. Alazab, "Deep learning techniques for behavioural malware analysis in cloud iaas," in *Malware Analysis Using Artificial Intelligence and Deep Learning*. Springer, 2021.

[10] S. Wali and I. Khan, "Explainable AI and random forest based reliable intrusion detection system," Tech. Rep., 2021.

[11] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," *IEEE Access*, vol. 8, pp. 73127–73141, 2020.

[12] *Cyber Security Operations Center (CSOC)*, Raytheon, 2017.

[13] J. P. Anderson, "Computer security threat monitoring and surveillance," Tech. Rep., 1980.

[14] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-2, no. 2, pp. 222–232, Feb. 1987.

[15] R. G. Bace and P. Mell, "Intrusion detection systems," Tech. Rep., 2001.

[16] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1–35, Jan. 2010.

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

[18] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proc. IEEE Symp. Secur. Privacy*, May 1999, pp. 120–132.

[19] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2015.

[20] M. Belouch, S. El Hadaj, and M. Idhammad, "Performance evaluation of intrusion detection based on machine learning using apache spark," *Proc. Comput. Sci.*, vol. 127, pp. 1–6, Jan. 2018.

[21] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: An overview," in *Proc. Int. Res. Conf. Eng. Technol. (IRCET)*, 2016.

[22] K. Kim and M. E. Aminanto, "Deep learning in intrusion detection perspective: Overview and further challenges," in *Proc. Int. Workshop Big Data Inf. Secur. (IWBIS)*, Sep. 2017, pp. 5–10.

[23] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu, "Explainable AI: A brief survey on history, research areas, approaches and challenges," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Springer, 2019, pp. 563–574.

[24] D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Mag.*, vol. 40, no. 2, pp. 44–58, Jun. 2019.

[25] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," 2017, *arXiv:1708.08296*.

[26] M. Alaa, "Artificial intelligence: Explainability, ethical issues and bias," *Ann. Robot. Autom.*, vol. 5, no. 1, pp. 34–37, Aug. 2021.

[27] R. A. Berk and J. Bleich, "Statistical procedures for forecasting criminal behavior: A comparative assessment," *Criminol. Pub. Pol'y*, vol. 12, p. 513, Jun. 2013.

[28] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "GEE: A gradient-based explainable variational autoencoder for network anomaly detection," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Jun. 2019, pp. 91–99.

[29] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue*, vol. 16, no. 3, pp. 31–57, Jun. 2018.

[30] M. V. Lent, W. Fisher, and M. Mancuso, "An explainable artificial intelligence system for small-unit tactical behavior," in *Proc. Nat. Conf. Artif. Intell.* Menlo Park, CA, USA: MIT Press, 2004, pp. 900–907.

[31] *Broad Agency Announcement Explainable Artificial Intelligence (XAI)*, DARPA-BAA-16–53, DARPA, 2016, pp. 7–8.

[32] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[33] K. Amarasinghe and M. Manic, "Improving user trust on deep neural networks based intrusion detection systems," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 3262–3268.

[34] J. Haspiel, N. Du, J. Meyerson, L. P. Robert, D. Tilbury, X. J. Yang, and A. K. Pradhan, "Explanations and expectations: Trust building in automated vehicles," in *Proc. Companion ACM/IEEE Int. Conf. Hum.-Robot Interact.*, Mar. 2018, pp. 119–120.

[35] M. P. S. Lorente, E. M. Lopez, L. A. Florez, A. L. Espino, J. A. I. Martínez, and A. S. de Miguel, "Explaining deep learning-based driver models," *Appl. Sci.*, vol. 11, no. 8, p. 3321, Apr. 2021.

[36] Y. Li, H. Wang, L. M. Dang, T. N. Nguyen, D. Han, A. Lee, I. Jang, and H. Moon, "A deep learning-based hybrid framework for object detection and recognition in autonomous driving," *IEEE Access*, vol. 8, pp. 194228–194239, 2020.

[37] J. Martinez-Cebrian, M.-A. Fernandez-Torres, and F. Diaz-De-Maria, "Interpretable global-local dynamics for the prediction of eye fixations in autonomous driving scenarios," *IEEE Access*, vol. 8, pp. 217068–217085, 2020.

[38] T. Ponn, T. Kröger, and F. Diermeyer, "Identification and explanation of challenging conditions for camera-based object detection of automated vehicles," *Sensors*, vol. 20, no. 13, p. 3699, Jul. 2020.

[39] A. Deeks, "The judicial demand for explainable artificial intelligence," *Columbia Law Rev.*, vol. 119, no. 7, pp. 1829–1850, 2019.

[40] O. Loyola-González, "Understanding the criminal behavior in Mexico City through an explainable artificial intelligence model," in *Proc. Mexican Int. Conf. Artif. Intell.* Springer, 2019, pp. 136–149.

[41] Q. Zhong, X. Fan, X. Luo, and F. Toni, "An explainable multi-attribute decision model based on argumentation," *Expert Syst. Appl.*, vol. 117, pp. 42–61, Mar. 2019.

[42] C. S. Vlek, H. Prakken, S. Renooij, and B. Verheij, "A method for explaining Bayesian networks for legal evidence with scenarios," *Artif. Intell. Law*, vol. 24, no. 3, pp. 285–324, Sep. 2016.

[43] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, "What do we need to build explainable AI systems for the medical domain?" 2017, *arXiv:1712.09923*.

[44] K. Gade, S. C. Geyik, K. Kenthapadi, V. Mithal, and A. Taly, "Explainable AI in industry," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 3203–3204.

[45] S. Itani, F. Lecron, and P. Fortemps, "A one-class classification decision tree based on kernel density estimation," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106250.

[46] L. Lindsay, S. Coleman, D. Kerr, B. Taylor, and A. Moorhead, "Explainable artificial intelligence for falls prediction," in *Proc. Int. Conf. Adv. Comput. Data Sci.* Springer, 2020, pp. 76–84.

[47] E. Pintelas, M. Liaskos, I. E. Livieris, S. Kotsiantis, and P. Pintelas, "Explainable machine learning framework for image classification problems: Case study on glioma cancer prediction," *J. Imag.*, vol. 6, no. 6, p. 37, May 2020.

[48] E. Prifti, Y. Chevaleyre, B. Hanczar, E. Belda, A. Danchin, K. Clément, and J.-D. Zucker, "Interpretable and accurate prediction models for metagenomics data," *GigaScience*, vol. 9, no. 3, Mar. 2020.

[49] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K. W. Low, S. F. Newman, J. Kim, and S. I. Lee, "Explainable machine learning predictions to help anesthesiologists prevent hypoxemia during surgery," *BioRxiv*, Jan. 2017, Art. no. 206540.

[50] L.-C. Huang, W. Yeung, Y. Wang, H. Cheng, A. Venkat, S. Li, P. Ma, K. Rasheed, and N. Kannan, "Quantitative structure–mutation–activity relationship tests (QSMART) model for protein kinase inhibitor response prediction," *BMC Bioinf.*, vol. 21, no. 1, pp. 1–22, Dec. 2020.

[51] A. Anguita-Ruiz, A. Segura-Delgado, R. Alcalá, C. M. Aguilera, and J. Alcalá-Fdez, "EXplainable artificial intelligence (XAI) for the identification of biologically relevant gene expression patterns in longitudinal human studies, insights from obesity research," *PLOS Comput. Biol.*, vol. 16, no. 4, Apr. 2020, Art. no. e1007792.

[52] S. M. Muddamsetty, M. N. Jahromi, and T. B. Moeslund, "Expert level evaluations for explainable ai (XAI) methods in the medical domain," in *Proc. Int. Conf. Pattern Recognit.* Springer, 2021, pp. 35–46.

[53] M. Graziani, V. Andrearczyk, S. Marchand-Maillet, and H. Müller, "Concept attribution: Explaining CNN decisions to physicians," *Comput. Biol. Med.*, vol. 123, Aug. 2020, Art. no. 103865.

[54] I. Rio-Torto, K. Fernandes, and L. F. Teixeira, "Understanding the decisions of CNNs: An in-model approach," *Pattern Recognit. Lett.*, vol. 133, pp. 373–380, May 2020.

[55] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[56] Y. E. Chun, S. B. Kim, J. Y. Lee, and J. H. Woo, "Study on credit rating model using explainable AI," *J. Korean Data Inf. Sci. Soc.*, vol. 32, no. 2, pp. 283–295, Mar. 2021.

[57] M. Han and J. Kim, "Joint banknote recognition and counterfeit detection using explainable artificial intelligence," *Sensors*, vol. 19, no. 16, p. 3607, Aug. 2019.

[58] E. Amparore, A. Perotti, and P. Bajardi, "To trust or not to trust an explanation: Using LEAF to evaluate local linear XAI methods," *PeerJ Comput. Sci.*, vol. 7, p. e479, Apr. 2021.

[59] J. van der Waa, E. Nieuwburg, A. Cremers, and M. Neerincx, "Evaluating XAI: A comparison of rule-based and example-based explanations," *Artif. Intell.*, vol. 291, Feb. 2021, Art. no. 103404.

[60] K. Sokol and P. Flach, "Explainability fact sheets: A framework for systematic assessment of explainable approaches," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 56–67.

[61] T. Rutkowski, K. Łapa, and R. Nielek, "On explainable fuzzy recommenders and their performance evaluation," *Int. J. Appl. Math. Comput. Sci.*, vol. 29, no. 3, pp. 595–610, Sep. 2019.

[62] X. Wang, D. Wang, and C. Xu, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Aug. 2019, pp. 5329–5336.

[63] G. Zhao, H. Fu, R. Song, T. Sakai, Z. Chen, X. Xie, and X. Qian, "Personalized reason generation for explainable song recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 4, pp. 1–21, Jul. 2019.

[64] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[65] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, 2018.

[66] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019.

[67] K. Gade, S. Geyik, K. Kenthapadi, V. Mithal, and A. Taly, "Explainable AI in industry: Practical challenges and lessons learned," in *Proc. Companion Proc. Web Conf.*, Apr. 2020, pp. 303–304.

[68] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," *Inf. Fusion*, vol. 76, pp. 89–106, Dec. 2021.

[69] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (XAI): A survey," 2020, *arXiv:2006.11371*.

[70] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, Dec. 2020.

[71] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should i trust you?' explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[72] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–9.

[73] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, ''Distribution-free predictive inference for regression,'' *J. Amer. Stat. Assoc.*, vol. 113, no. 523, pp. 1094–1111, Jul. 2018.

[74] A. Shrikumar, P. Greenside, and A. Kundaje, ''Learning important features through propagating activation differences,'' in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.

[75] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, ''Layer-wise relevance propagation for neural networks with local renormalization layers,'' in *Proc. Int. Conf. Artif. Neural Netw.* Springer, 2016, pp. 63–71.

[76] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, and F. Viegas, ''Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV),'' in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2668–2677.

[77] C. Yang, A. Rangarajan, and S. Ranka, ''Global model interpretation via recursive partitioning,'' in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Communications; IEEE 16th Int. Conf. Smart City; IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1563–1570.

[78] M. A. Valenzuela-Escárcega, A. Nagesh, and M. Surdeanu, ''Lightly-supervised representation learning with global interpretability,'' 2018, *arXiv:1805.11545*.

[79] A. Rosenfeld and A. Richardson, ''Explainability in human–agent systems,'' *Auto. Agents Multi-Agent Syst.*, vol. 33, no. 6, pp. 673–705, Nov. 2019.

[80] V. Arya, R. K. E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović, S. Mourad, P. Pedemonte, R. Raghavendra, J. Richards, P. Sattigeri, K. Shanmugam, M. Singh, K. R. Varshney, D. Wei, and Y. Zhang, ''One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques,'' 2019, *arXiv:1909.03012*.

[81] F. Doshi-Velez and B. Kim, ''Towards a rigorous science of interpretable machine learning,'' 2017, *arXiv:1702.08608*.

[82] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, ''Explaining explanations: An overview of interpretability of machine learning,'' in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2018, pp. 80–89.

[83] D. A. Melis and T. Jaakkola, ''Towards robust interpretability with self-explaining neural networks,'' *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.

[84] R. Luss, P.-Y. Chen, A. Dhurandhar, P. Sattigeri, K. Shanmugam, and C.-C. Tu, ''Generating contrastive explanations with monotonic attribute functions,'' Tech. Rep., 2019.

[85] I. Butun, S. D. Morgera, and R. Sankar, ''A survey of intrusion detection systems in wireless sensor networks,'' *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 1st Quart., 2014.

[86] A. Sharma and S. K. Sahay, ''Evolution and detection of polymorphic and metamorphic malwares: A survey,'' 2014, *arXiv:1406.7061*.

[87] V. Chandola, A. Banerjee, and V. Kumar, ''Anomaly detection: A survey,'' *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[88] O. Loyola-Gonzalez, ''Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view,'' *IEEE Access*, vol. 7, pp. 154096–154113, 2019.

[89] J. H. Friedman, ''Greedy function approximation: A gradient boosting machine,'' *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[90] D. W. Apley and J. Zhu, ''Visualizing the effects of predictor variables in black box supervised learning models,'' *J. Roy. Stat. Soc., B, Stat. Methodol.*, vol. 82, no. 4, pp. 1059–1086, Sep. 2020.

[91] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, ''Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation,'' *J. Comput. Graph. Statist.*, vol. 24, no. 1, pp. 44–65, 2015.

[92] J. H. Friedman and B. E. Popescu, ''Predictive learning via rule ensembles,'' *Ann. Appl. Statist.*, vol. 2, no. 3, pp. 916–954, Sep. 2008.

[93] V. Petsiuk, A. Das, and K. Saenko, ''RISE: Randomized input sampling for explanation of black-box models,'' 2018, *arXiv:1806.07421*.

[94] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, ''Not just a black box: Learning important features through propagating activation differences,'' 2016, *arXiv:1605.01713*.

[95] M. Sundararajan, A. Taly, and Q. Yan, ''Axiomatic attribution for deep networks,'' in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

[96] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ''Grad-CAM: Visual explanations from deep networks via gradient-based localization,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[97] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, ''Explaining nonlinear classification decisions with deep Taylor decomposition,'' *Pattern Recognit.*, vol. 65, pp. 211–222, May 2017.

[98] S. R. Islam, W. Eberle, S. K. Ghafoor, A. Siraj, and M. Rogers, ''Domain knowledge aided explainable artificial intelligence for intrusion detection and response,'' 2019, *arXiv:1911.09853*.

[99] M. Sarhan, S. Layeghy, and M. Portmann, ''An explainable machine learning-based network intrusion detection system for enabling generalisability in securing IoT networks,'' 2021, *arXiv:2104.07183*.

[100] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, ''Explaining anomalies detected by autoencoders using SHAP,'' 2019, *arXiv:1903.02407*.

[101] I. A. Khan, N. Moustafa, D. Pi, K. M. Sallam, A. Y. Zomaya, and B. Li, ''A new explainable deep learning framework for cyber threat discovery in industrial IoT networks,'' *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11604–11613, Jul. 2022.

[102] C. Wu, A. Qian, X. Dong, and Y. Zhang, ''Feature-oriented design of visual analytics system for interpretable deep learning based intrusion detection,'' in *Proc. Int. Symp. Theor. Aspects Softw. Eng. (TASE)*, Dec. 2020, pp. 73–80.

[103] N. Burkart, M. Franz, and M. F. Huber, ''Explanation framework for intrusion detection,'' in *Machine Learning for Cyber Physical Systems*. Berlin, Germany: Springer, 2021, pp. 83–91.

[104] K. Amarasinghe, K. Kenney, and M. Manic, ''Toward explainable deep neural network based anomaly detection,'' in *Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI)*, Jul. 2018, pp. 311–317.

[105] J. Kauffmann, K.-R. Müller, and G. Montavon, ''Towards explaining anomalies: A deep Taylor decomposition of one-class models,'' 2018, *arXiv:1805.06230*.

[106] M. Szczepanski, M. Choras, M. Pawlicki, and R. Kozik, ''Achieving explainability of intrusion detection system by hybrid oracle-explainer approach,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[107] G. Pang, C. Ding, C. Shen, and A. van den Hengel, ''Explainable deep few-shot anomaly detection with deviation networks,'' 2021, *arXiv:2108.00462*.

[108] Q.-V. Dang, ''Understanding the decision of machine learning based intrusion detection systems,'' in *Proc. Int. Conf. Future Data Secur. Eng.* Springer, 2020, pp. 379–396.

[109] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, ''Explaining recurrent neural network predictions in sentiment analysis,'' 2017, *arXiv:1706.07206*.

[110] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and A. K.-R. Müller, ''Layer-wise relevance propagation: An overview,'' in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. 2019, pp. 193–209.

[111] F. P. Caforio, G. Andresini, G. Vessio, A. Appice, and A. D. Malerba, ''Leveraging grad-CAM to improve the accuracy of network intrusion detection systems,'' in *Discovery Science*. 2021.

[112] J. Kauffmann, L. Ruff, G. Montavon, and K.-R. Müller, ''The clever Hans effect in anomaly detection,'' 2020, *arXiv:2006.10609*.

[113] L. K. Hansen and L. Rieger, ''Interpretability in intelligent systems—A new concept?'' in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 41–49.

[114] E. Pintelas, I. E. Livieris, and P. Pintelas, ''A grey-box ensemble model exploiting black-box accuracy and white-box intrinsic interpretability,'' *Algorithms*, vol. 13, no. 1, p. 17, Jan. 2020.

[115] S. R. Islam, W. Eberle, S. K. Ghafoor, and M. Ahmed, ''Explainable artificial intelligence approaches: A survey,'' 2021, *arXiv:2101.09429*.

[116] A. P. Kuruvila, X. Meng, S. Kundu, G. Pandey, and K. Basu, ''Explainable machine learning for intrusion detection via hardware performance counters,'' *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, Feb. 7, 2022, doi: 10.1109/TCAD.2022.3149745.

[117] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, ''Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model,'' *Complexity*, vol. 2021, pp. 1–11, Jan. 2021.

[118] T. Dias, N. Oliveira, N. Sousa, I. Praça, and O. Sousa, ''A hybrid approach for an interpretable and explainable intrusion detection system,'' 2021, *arXiv:2111.10280*.

[119] C. F. T. Pontes, M. M. C. de Souza, J. J. C. Gondim, M. Bishop, and M. A. Marotta, ''A new method for flow-based network intrusion detection using the inverse Potts model,'' *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1125–1136, Jun. 2021.

[120] C. S. Wickramasinghe, K. Amarasinghe, D. L. Marino, C. Rieger, and M. Manic, "Explainable unsupervised machine learning for cyber-physical systems," *IEEE Access*, vol. 9, pp. 131824–131843, 2021.

[121] C. Langin, M. Wainer, and S. Rahimi, "ANNaBell Island: A 3D color hexagonal SOM for visual intrusion detection," *Int. J. Comput. Sci. Inf. Secur.*, vol. 9, no. 1, pp. 1–7, 2011.

[122] B. Subba, S. Biswas, and S. Karmakar, "Intrusion detection systems using linear discriminant analysis and logistic regression," in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2015, pp. 1–6.

[123] Z. Wang, J. Yang, and F. Li, "A new anomaly detection method based on IGTE and IGFE," in *Proc. Int. Conf. Secur. Privacy Commun. Netw.* Springer, 2014, pp. 93–109.

[124] Z. Wang, J. Yang, Z. ShiZe, and C. Li, "Robust regression for anomaly detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[125] O. Loyola-Gonzalez, A. E. Gutierrez-Rodriguez, M. A. Medina-Perez, R. Monroy, J. F. Martinez-Trinidad, J. A. Carrasco-Ochoa, and M. Garcia-Borroto, "An explainable artificial intelligence model for clustering numerical databases," *IEEE Access*, vol. 8, pp. 52370–52384, 2020.

[126] N. Frost, M. Moshkovitz, and C. Rashtchian, "ExKMC: Expanding explainable K-means clustering," 2020, *arXiv:2006.02399*.

[127] S. Dasgupta, N. Frost, M. Moshkovitz, and C. Rashtchian, "Explainable K-means and K-medians clustering," in *Proc. 37th Int. Conf. Mach. Learn.*, Vienna, Austria, 2020, pp. 12–18.

[128] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," in *Proc. 15th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 1999, pp. 371–377.

[129] A. Ojugo, A. Eboka, O. Okonta, R. Yoro, and F. Aghware, "Genetic algorithm rule-based intrusion detection system (GAIDS)," *J. Emerg. Trends Comput. Inf. Sci.*, vol. 3, no. 8, pp. 1182–1194, 2012.

[130] K. Chadha and S. Jain, "Hybrid genetic fuzzy rule based inference engine to detect intrusion in networks," in *Intelligent Distributed Computing*. Springer, 2015, pp. 185–198.

[131] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proc. LISA*, vol. 99, 1999, pp. 229–238.

[132] B. Caswell, J. Beale, and A. Baker, "Snort intrusion detection prevention toolkit," Syngress, Tech. Rep., 2007.

[133] A. Qayyum, M. H. Islam, and M. Jamil, "Taxonomy of statistical based anomaly detection techniques for intrusion detection," in *Proc. IEEE Symp. Emerg. Technol.*, Sep. 2005, pp. 270–276.

[134] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019.

[135] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, 1st Quart., 2013.

[136] A. T. Tran, "Network anomaly detection," in *Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM) Focal Topic: Advanced Persistent Threats*. 2017.

[137] A. B. Ashfaq, M. Javed, S. A. Khayam, and H. Radha, "An information-theoretic combining method for multi-classifier anomaly detection systems," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.

[138] W. Sha, Y. Zhu, T. Huang, M. Qiu, Y. Zhu, and Q. Zhang, "A multi-order Markov chain based scheme for anomaly detection," in *Proc. IEEE 37th Annu. Comput. Softw. Appl. Conf. Workshops*, Jul. 2013, pp. 83–88.

[139] N. Ye, "A Markov chain model of temporal behavior for anomaly detection," in *Proc. IEEE Syst., Man, Cybern. Inf. Assurance Secur. Workshop*, vol. 166, Jun. 2000, p. 169.

[140] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proc. IEEE*, vol. 84, no. 10, pp. 1358–1384, Oct. 1996.

[141] A. J. Hoglund, K. Hatonen, and A. S. Sorvari, "A computer host-based user anomaly detection system using the self-organizing map," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. Neural Comput., New Challenges Perspect. New Millennium*, 2000, pp. 411–416.

[142] P. Lichodzijewski, A. N. Zincir-Heywood, and M. I. Heywood, "Host-based intrusion detection using self-organizing maps," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2002, pp. 1714–1719.

[143] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "A hierarchical SOM-based intrusion detection system," *Eng. Appl. Artif. Intell.*, vol. 20, no. 4, pp. 439–451, 2007.

[144] C. Langin, H. Zhou, and S. Rahimi, "A model to use denied internet traffic to indirectly discover internal network security problems," in *Proc. IEEE Int. Perform., Comput. Commun. Conf.*, Dec. 2008, pp. 486–490.

[145] C. Langin, H. Zhou, S. Rahimi, B. Gupta, M. Zargham, and M. R. Sayeh, "A self-organizing map and its modeling for discovering malignant network traffic," in *Proc. IEEE Symp. Comput. Intell. Cyber Secur.*, Mar. 2009, pp. 122–129.

[146] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 313–316.

[147] M. Sölch, "Detecting anomalies in robot time series data using stochastic recurrent networks," Tech. Rep., 2015.

[148] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[149] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 218–224.

[150] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228.

[151] A. Holzinger, "From machine learning to explainable AI," in *Proc. World Symp. Digit. Intell. Syst. Mach. (DISA)*, Aug. 2018, pp. 55–66.

[152] H. Liu, C. Zhong, A. Alnusair, and S. R. Islam, "FAIXID: A framework for enhancing AI explainability of intrusion detection results using data cleaning techniques," *J. Netw. Syst. Manage.*, vol. 29, no. 4, pp. 1–30, Oct. 2021.

[153] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[154] (1999). *KDD Cup 1999 Data the UCI KDD Archive*. Accessed: Apr. 9, 2022. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[155] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, vol. 1, Jan. 2018, pp. 108–116.

[156] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "UGR-16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Comput. Secur.*, vol. 73, pp. 411–424, Mar. 2018.

[157] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Proc. Sci. Inf. Conf.*, Aug. 2014, pp. 372–378.

[158] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[159] F. R. S. K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Philosoph. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.

[160] *Shapash Welcome to Shapash's Documentation*. Accessed: Jul. 23, 2022. [Online]. Available: https://shapash.readthedocs.io/en/latest/overview.html

[161] O. Dijk. (2019). *ExplainerDashboard Starting the Default Dashboard*. Accessed: Jul. 22, 2022. [Online]. Available: https://explainerdashboard.readthedocs.io/en/latest/dashboards.html

[162] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: Adversarially-tuned domain generation and detection," in *Proc. ACM Workshop Artif. Intell. Secur.*, 2016, pp. 13–21.

[163] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017, *arXiv:1712.05526*.

[164] N. Rastogi, S. Rampazzi, M. Clifford, M. Heller, M. Bishop, and K. Levitt, "Explaining RADAR features for detecting spoofing attacks in connected autonomous vehicles," 2022, *arXiv:2203.00150*.

[165] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, and X. Yin, "Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2632–2647, Aug. 2021.

[166] M. Pawlicki, M. Choraś, and R. Kozik, "Defending network intrusion detection systems against adversarial evasion attacks," *Future Gener. Comput. Syst.*, vol. 110, pp. 148–154, Sep. 2020.

[167] A. Hartl, M. Bachl, J. Fabini, and T. Zseby, "Explainability and adversarial robustness for RNNs," in *Proc. IEEE 6th Int. Conf. Big Data Comput. Service Appl. (BigDataService)*, Aug. 2020, pp. 148–156.

SUBASH NEUPANE received the bachelor's degree in computer engineering from Kathmandu University, Nepal, in 2011, the M.S. degree in information technology (professional computing) from the Swinburne University of Technology, Melbourne, Australia, in 2016, and the M.S. degree in information systems and security management from Tuskegee University, AL, USA, in 2020. He is currently pursuing the Ph.D. degree in systems and security with Mississippi State University. He was a Telecom Network Engineer with Tandem Corporation, Melbourne, before moving to the USA. His current research interests include systems and security, machine learning, and blockchain.

JESSE ABLES (Graduate Student Member, IEEE) received the B.S. degree in software engineering, in 2016, and the M.S. degree in cyber security, in 2019. He is currently pursuing the Ph.D. degree in autonomous cyber security with Mississippi State University (MSU). He worked as a Teacher in computer science field at MSU and as an English Teacher in South Korea. He is currently working as a Research Assistant with the Computer Science and Engineering Department, MSU. His research interests include autonomous security, the Internet of Things, and anomaly detection.

WILLIAM ANDERSON received the B.S. degree in computer science from Mississippi State University, in 2019, where he is currently pursuing the M.S. degree in artificial intelligence. In 2020, he worked as a Research Assistant at NSPARC, developing natural language processing applications for workforce development. He is also working as a Research Assistant with the Computer Science and Engineering Department, Mississippi State University. His research interests include natural language processing, anomaly detection in the fields of cybersecurity and healthcare, and explainable AI.

SUDIP MITTAL (Member, IEEE) received the Ph.D. degree in computer science from the University of Maryland Baltimore County, in 2019. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Mississippi State University. His primary research interests include cybersecurity and artificial intelligence. His goal is to develop the next generation of cyber defense systems that help protect various organizations and people. At Mississippi State University, he leads the Secure and Trustworthy Cyberspace (SECRETS) Laboratory and has published over 60 journals and conference papers in leading cybersecurity and AI venues. He has received funding from the NSF, USAF, USACE, and various other the Department of Defense Programs. He also serves as a program committee member or the program chair of leading AI and cybersecurity conferences and workshops. His work has been cited in the LA Times, Business Insider, WIRED, and Cyberwire. He is a member of the ACM.

SHAHRAM RAHIMI (Member, IEEE) is currently a Professor and the Head of the Department of Computer Science and Engineering, Mississippi State University. Prior to that, he led the Department of Computer Science, Southern Illinois University, for five years. He is also a Recognized Leader in the area of artificial and computational intelligence with over 220 peer-reviewed publications and a few patents or pending patents in this area. He has served as the Editor-in-Chief for two leading *Computational Intelligence* journals and sits on the editorial board of several others. He is a member of IEEE New Standards Committee in Computational Intelligence and provides advice to staff and administration at federal government on predictive analytics for foreign policy. He was a recipient of 2016 Illinois Rising Star Award from ISBA, selected among 100's of other highly qualified candidates. His intelligent algorithm for patient flow optimization and hospital staffing is currently used in over 1000 emergency departments across the nation and was named top ten AI technology for healthcare, in 2018, by *HealthTech Magazine*. He has secured over $20M of federal and industry funding as a PI or a co-PI in the last 20 years. He has also organized 15 conferences and workshops in the areas of computational intelligence and multi-agent systems over the past two decades.

IOANA BANICESCU (Life Senior Member, IEEE) received the Diploma degree in engineering (electronics and telecommunications) from the Polytechnic University of Bucharest and the M.S. and Ph.D. degrees in computer science from New York University—Polytechnic Institute. Between 2009 and 2017, she was the Director of the Center for Cloud and Autonomic Computing, Mississippi State University (MSU), and also the Co-Director of the National Science Foundation Center for Cloud and Autonomic Computing. She is currently a Professor with the Department of Computer Science and Engineering, MSU. Her research interests include parallel algorithms, scientific computing, scheduling theory, load balancing algorithms, performance modeling, analysis and prediction, autonomic computing, performance optimization for problems in computational science, and graph analytics. She has given many invited talks at universities, government laboratories, and at various national and international forums in the USA and overseas. She was a recipient of a number of awards for research and scholarship from the National Science Foundation (NSF). She served and continues to serve on numerous research review panels for advanced research grants in the USA and Europe, on steering and program committees of a number of international ACM and IEEE conferences, symposia, and workshops, and on the Executive Board and the Advisory Board of the IEEE Technical Committee on Parallel Processing (TCPP). She was an Associate Editor of the *Cluster Computing* journal and the *International Journal on Computational Science and Engineering*. Over the years, she was recognized with many distinctions for her scholarly contributions.

MARIA SEALE received the B.S. degree in computer science from the University of Southern Mississippi and the M.S. and Ph.D. degrees in computer science from Tulane University. She is currently a Computer Scientist at the Information Technology Laboratory, U.S. Army Engineer Research and Development Center (ERDC). She has over 20 years of experience in research, development, and teaching in computer science. She has held positions at the Institute for Naval Oceanography, the U.S. Naval Research Laboratory, and various private companies, and a tenured associate professorship at the University of Southern Mississippi. Her experience has included work with ocean modeling, underwater seismic data collection and processing, geographical information systems design, natural language processing, and machine learning. At ERDC, she has been involved with research in making scalable machine learning algorithms available on high performance computing platforms and expanding the laboratory's capabilities to manage and analyze very large data sets.