Annals of Telecommunications (2022) 77:789–812
https://doi.org/10.1007/s12243-022-00926-7

Explainable artificial intelligence for cybersecurity: a literature survey

Fabien Charmet[1] · Harry Chandra Tanuwidjaja[1] · Solayman Ayoubi[2] · Pierre-François Gimenez[3] · Yufei Han[4] · Houda Jmila[5] · Gregory Blanc[5] · Takeshi Takahashi[1] · Zonghua Zhang[6]

Received: 30 June 2022 / Accepted: 31 August 2022 / Published online: 26 October 2022

Abstract
With the extensive application of deep learning (DL) algorithms in recent years, e.g., for detecting Android malware or vulnerable source code, artificial intelligence (AI) and machine learning (ML) are increasingly becoming essential in the development of cybersecurity solutions. However, sharing the same fundamental limitation with other DL application domains, such as computer vision (CV) and natural language processing (NLP), AI-based cybersecurity solutions are incapable of justifying the results (ranging from detection and prediction to reasoning and decision-making) and making them understandable to humans. Consequently, explainable AI (XAI) has emerged as a paramount topic addressing the related challenges of making AI models explainable or interpretable to human users. It is particularly relevant in cybersecurity domain, in that XAI may allow security operators, who are overwhelmed with tens of thousands of security alerts per day (most of which are false positives), to better assess the potential threats and reduce alert fatigue. We conduct an extensive literature review on the intersection between XAI and cybersecurity. Particularly, we investigate the existing literature from two perspectives: the applications of XAI to cybersecurity (e.g., intrusion detection, malware classification), and the security of XAI (e.g., attacks on XAI pipelines, potential countermeasures). We characterize the security of XAI with several security properties that have been discussed in the literature. We also formulate open questions that are either unanswered or insufficiently addressed in the literature, and discuss future directions of research.

Keywords Cybersecurity · Explainable AI · Machine learning

Solayman Ayoubi, Pierre-François Gimenez, Yufei Han and Houda Jmila contributed equally to this work.

✉ Fabien Charmet
fabien.charmet@nict.go.jp

Harry Chandra Tanuwidjaja
harry@nict.go.jp

Solayman Ayoubi
solayman.ayoubi@loria.fr

Pierre-François Gimenez
pierre-francois.gimenez@centralesupelec.fr

Yufei Han
yufei.han@inria.fr

Houda Jmila
houda.jmila@telecom-sudparis.eu

Gregory Blanc
gregory.blanc@telecom-sudparis.eu

Takeshi Takahashi
takeshi_takahashi@ieee.org

Zonghua Zhang
zonghua.zhang@ieee.org

[1] National Institute of Information and Communications Technology, 4-2-1 Nukuikitamachi Koganei, Tokyo 184-8795, Japan

[2] LORIA, Universite de Lorraine, Lorraine, France

[3] CentraleSupélec, IRISA, University Rennes, Rennes, France

[4] Inria, IRISA, University Rennes, Rennes, France

[5] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, France

[6] Huawei Paris Research Center, Paris, France

1 Introduction

Artificial intelligence (AI) is a paradigm for simulating human reasoning, e.g., classifying previously unobserved data, predicting future events such as stock market trends, forecasting sales and consumer behavior. The goal of this field has been to develop general intelligence in AI [23]. Interest in AI-related research has been growing exponentially, partly motivated by its outstanding performance in computer vision (CV) and speech recognition. As researchers focus on improving model performance, the ability to explain the reasoning behind a model's predictions becomes increasingly crucial. For example, "Why did I not get a loan?" and "Why does this X-ray picture say I have cancer?" are compelling questions that the research community must be able to answer. Therefore, researchers have been exploring explainable AI (XAI), which is a paradigm that targets AI models and aims to provide explanations for their predictions.

Researchers have discussed the need to provide explanations for their models regarding several practical [44, 135], ethical [50], and operational [77] considerations. DARPA's XAI program [44] highlighted that a machine learning (ML) model's explainability is inversely proportional to its prediction performance (e.g., accuracy). Notably, deep learning (DL) models, which are arguably the most robust and complex type of AI algorithms, are also the most difficult to explain. The role of XAI is to enhance explainability while maintaining high performance. Xu et al. [135] claimed that XAI is essential for i) professionals (e.g., doctors) using AI systems to understand the decisions made, ii) end users (e.g., patients) who are affected by an AI decision—there are legal regulations that codify this need, such as the General Data Protection Regulation (GDPR) [1], and iii) developers to improve AI algorithms by accurately identifying their strengths and weaknesses.

Holzinger et al. [50] highlighted the intersection between security and explainable ML. They argued that XAI could be used to select the right data anonymization techniques so that privacy is protected while the ML results remain viable. To comply with the GDPR [1], researchers resort to anonymizing data they use. However, several standard anonymization techniques distort the predictions of ML algorithms. Researchers suggest that AI explainability could help in selecting ideal anonymization techniques for ML algorithms, as comprehending the ML decisions would aid in understanding and estimating bias. Thus, XAI could be the key to designing solutions that leverage the power of ML while protecting privacy. There are two main approaches to explain deep neural networks (DNNs): i) making parts of a DNN transparent—sensitivity analysis [104] and layer-wise relevance propagation (LRP) [15] are well-known methods, with superior performance for LRP to identify the most relevant pixels; ii) learning semantic graphs called explanatory graphs from existing DNNs, which aim to extract the knowledge learned by a DNN and model it as an explainable graph, as proposed by Zhang et al. [138].

Longo et al. [77] classified studies on XAI into two approaches: a minority of works that focuses on creating inherently explainable models, and the majority that wraps black-box models with a layer of explainability, the so-called post hoc models. They also argued that explainability might be more attractive in some domains than others, including critical domains such as threat detection, protection against adversarial attacks, physician decision support, autonomous vehicles, and object detection. These domains are frequently explained by saliency maps [51]. From the perspective of technical challenges, they highlighted several issues, including the lack of a common approach to evaluate and compare AI models and the need to interpret explanations in the form of visualization or human-readable text. We conducted a rigorous literature review by investigating relevant papers from eight major digital academic libraries: Google Scholar, IEEE Xplore, ScienceDirect, SpringerLink, ACM Digital Library, Scopus, ResearchGate, and Semantic Scholar. During the collection process, we combed these libraries based on a keyword search as follows: i) explanation-related terms such as "explainable," "interpretable," "understandable," "intelligible," and "comprehensible"; ii) AI-related terms, including "AI," "XAI," "ML," "DL," "classification," and "prediction" and "black-box."; and iii) security-related terms, such as "adversary," "threats," "attack," "cybersecurity," and "detection."

To understand recent advancements in the field, the search was limited to publications between 2000 and 2022. The collated papers were classified by topic, i.e., applying XAI to cybersecurity or the security of XAI methods. We also checked the reference lists of the selected papers and employed a cascading strategy to identify additional papers, yielding a final list of 50 papers. The XAI has been broadly surveyed in several studies [2, 29, 47, 105]. Because it is an emerging field, the existing literature can be justifiably surveyed without a specific domain scope. However, the recent advancements and increasing threats in the real world warrant a cybersecurity review of XAI. There are two existing works that address this topic from a high-level perspective: [6] and [121]. In [6], Mohiuddin et al. discussed the topic of XAI through the lens of multiple applications: healthcare, smart cities, NLP, security, etc. From a cybersecurity perspective, most of their analyses targeted intrusion detection systems, and their usage in the previously identified applications. The works discussed in the various cybersecurity subsections of that book did not explore the technical considerations and omitted several implementation details and technical results. Similarly, [121] considered various applications for AI models, but always presented a cybersecurity approach (as opposed to the scope of [6]). The analysis was divided into three questions: what is the motivation for applying AI to a specific domain; what are the technical requirements for it; and how can XAI help with achieving the goals presented in the motivation?

We argue that this work is distinct from the existing literature in two main aspects. First, this survey focuses on how to apply explainability and which methods are relevant to cybersecurity applications. Second, this survey analyses the security of XAI methods and identifies existing trends and challenges. It outlines technical research avenues that would immensely contribute to the AI and XAI research community.

The contributions of this survey are as follows:

- We provide a comprehensive background with the main concepts, existing methods, limitations, and risks associated with securing explainable systems.
- We collect and analyze 50 papers and organize them in a cybersecurity-oriented taxonomy.
- We discuss open research problems and identified multiple research avenues for future work.

The remainder of this article is organized as follows. Section 2 introduces the taxonomy of XAI and scope of this survey. Section 3 introduces the surveyed terms, models, and XAI methods. Section 4 reviews the state of the art of explainable classifiers for cybersecurity tasks. Section 5 explores the security of XAI methods. In Section 6, we present unaddressed AI-related research questions and our perspective on the future of XAI in cybersecurity. Finally, we conclude this survey in Section 7.

2 Taxonomy

AI models are a major actor in the cybersecurity research landscape. However, ensuring the proper use of AI models in a cybersecurity context is an arduous task. Shaukat et al. [111] provide a broad review on the applications of ML techniques to cybersecurity. Cybersecurity applications of AI encompass network security, computer security, mobile security, etc. In essence, AI and XAI methods have been implemented on various datasets corresponding to the research trends. As such, we believe that discussing the broadest range of applications will give the reader a diversified vision about the landscape of XAI in a cybersecurity context. Therefore, we argue that transparency of and trust in AI also belong to the scope of cybersecurity as they contribute to reducing the potential maliciousness toward the AI model and the system in general. We extend this approach to the analysis of the security of XAI methods, where we discuss various works either compromising the explanations or defending them against unwanted/unexpected behaviors.

XAI is a growing research domain, to which researchers have contributed different definitions and perspectives owing to a lack of standardization. For example, the authors of [2, 128] employed the approach of the six W questions—What, Who, When, Why, Where, and How. This approach helped identify different stakeholders in AI-based systems and define the scope of XAI and the reasoning behind the need for XAI. Another approach was to characterize XAI through its intrinsic properties. Arrieta et al. [13] classified XAI models as white box or post hoc models, whereas the authors of [46] and [94] outlined desirable properties for XAI. Hagras [46] discussed the link between human-understandable information and the flexibility of the data labeling process. Paredes et al. [94] discussed explanations for cybersecurity and insisted that explanations should be able to capture changes in an attacker's strategy, or to help identify anomalies when they are outlined by detection mechanisms. Kuppa et al. [66] proposed a taxonomy for XAI concerning its security properties. They also demonstrated a novel black-box attack on explainable models and evaluated it on three datasets. The proposed taxonomy covered three domains: the explanations of predictions made by a model, the security properties associated with models (i.e., confidentiality, integrity, and privacy), and the threat models used. The authors differentiated confidentiality and privacy by highlighting that the former pertained to the features of data, while the latter pertained to the explanations given to various security actors. We employed a different approach by considering the intersection of XAI and cybersecurity (Table 1). In this survey, we explored both methods for explaining AI-based cybersecurity applications and security analyses of XAI methods. In the first case, the literature we surveyed covers various practical scenarios, mostly supported by cybersecurity datasets (23 papers). In the second case, we identified several properties with respect to the security of XAI that were discussed in the state of the art (27 papers). Our taxonomy differs from those of existing XAI surveys, as most of them considered XAI from an intrinsic perspective. Numerous explanation methods were not attacked; however, the security properties presented in this survey should be relevant for these methods as well. We do not mention these methods in this survey to preserve our cybersecurity perspective. Table 1 describes the classification of the existing literature regarding our taxonomy.

3 Preliminaries

Before discussing the intersection of XAI and cybersecurity, we remind the reader of certain terms, and present a few intrinsically explainable models and explainability methods. As mentioned in Section 2, we do not introduce the reader to models or explanation methods that were not encountered in the surveyed literature.

Table 1 Classification of the surveyed literature

			References
XAI & Cybersecurity	Explainable Classification for Cybersecurity	XAI for transparency and trust	[8, 11, 39, 45, 53, 54, 57, 58, 79, 106, 109, 123, 124, 132, 133, 143]
		XAI for improving the performances	[43, 102, 131]
		XAI for explaining errors	[32, 34, 76, 82]
	Cybersecurity of XAI methods	Fairness	[7, 10, 27, 66, 72, 99, 117, 118]
		Integrity	[20, 28, 37, 49, 66, 115, 139]
		Privacy	[65, 66, 113, 141]
		Confidentiality	[65, 66, 84]
		Robustness	[17, 38, 56, 59, 65, 66, 69, 70, 83, 112]
		Explanation evaluation	[3, 52, 74]

3.1 Glossary

In the literature, "explainable," "interpretable," and "understandable" have been used interchangeably. We agree that explainable and interpretable are synonyms, but "understandable" is not. We define the terms interpretable/explainable and understandable using the studies of Arrieta et al. [13] and Molnar [85].

Explainable/interpretable Explainability (interpretability) can be defined as the ability to provide the meaning of the relationships a model's inputs and its outcomes have, in a human-readable form [85]. In the XAI field, explainability (interpretability) is the degree to which the decision made by an AI model can be understood by humans. The higher the explainability (interpretability), the easier it is for humans to comprehend why a model made a decision.

Understandable Understandability can be defined as the capability of an AI model to make a human understand its function without needing to explain the model's intrinsic mechanisms [13]. In this survey, we discuss scientific contributions from the perspective of explainability (interpretability), as it is the most common approach in the literature. We argue that the literature is not mature and does not provide distinct definitions for these terms. We considered the philosophical issue of understandability to be outside the scope of this survey but provided a tentative definition to guide readers.

3.2 Explainable models

The models presented in this section can provide explanations without requiring an external XAI method. The intrinsic mechanisms of the model can be extracted, and the model can provide information in a human-understandable way.

Linear regression model (LR) [48] is a linear approach for modeling the relationship between feature inputs and their outcomes. An LR model linearly approximates results using the weighted sum of the feature inputs. The formula of LR is expressed as follows:

$$y = \beta_0 + \beta_1 \chi_1 + \beta_2 \chi_2 + \ \ldots \ + \beta_n \chi_n + \epsilon$$

where y represents the regression target (output), χ represents the feature input, β_n denotes the weight value, and ϵ represents the error term.

The logistic regression model [61] is an extension of LR because LR treats classes as numbers, 0 or 1, and attempts to find a hyperplane that minimizes the distance between points and the hyperplane. In other words, the LR model cannot efficiently solve classification problems. Logistic regression is not based on probability; rather, it is merely a simple interpolation process. Thus, linear interpolation cannot provide a meaningful threshold for distinguishing classes.

The generalized linear model (GLM) [33] is another extension of LR model. The GLM addresses the problem that a simple weighted sum in LR is too restrictive for real-world problems. LR requires the assumption that the target outcome follows a Gaussian distribution, whereas the GLM allows a non-Gaussian distribution and connects the weighted sum of distributions through a nonlinear function. The formula for the GLM can be described as follows:

$$g(E_y(y|x)) = \beta_0 + \beta_1 x_1 + + \beta_2 x_2 + \ \ldots \ + \beta_n x_n$$

where g denotes link function, $\beta_n x_n$ denotes the weighted sum, and E_y denotes the distribution probability.

Decision tree (DT) [60] is a graph representation of data instances. In summary, DT splits data multiple times based on defined cutoff values (decision nodes), thereby creating different subsets. Each instance will belong to a specific

subset. The excellence of the DT algorithm arises when the relationship between features and outcomes is nonlinear. The tree structure algorithm enables the processing of nonlinear data. The formula for a DT can be expressed as follows:

$$y = \sum_{m=1}^{M} c_m I\{x \in R_m\}$$

where x represents the input feature, y represents the output, $I\{x \in R_m\}$ denotes the identity function, and R_m denotes the leaf node.

The construction of a DT is a recursive splitting process. The algorithm attempts to create subsets by grouping all data points until the best partition (based on the information gain theory) has been identified. The DT model has a simple interpretation from the explainability perspective. From the root node, we go to the next node until the desired subset is found. Figure 1 illustrates a decision node. The visualization bolsters the explainability of the decision-making process, as it is based on if–then rules that start from the root node. If rule A is met, we proceed to decision node A. If not, we will go down to decision node B. We repeat this process until we reach the leaf node, which reveals the predicted outcome. In a DT, each feature has a "significance level," which is called *feature importance*. The overall model importance is at 100, which is then passed along all branches. This means that each feature has a share in the overall model importance. The prediction in a DT can be explained by the formula:

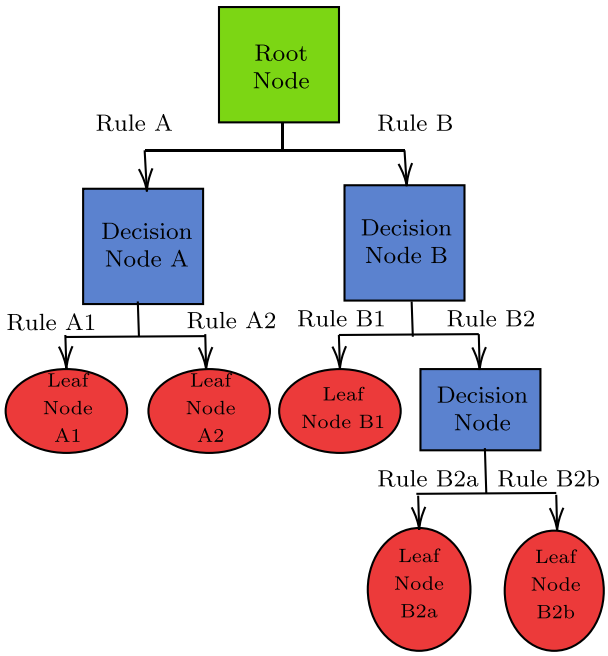

Fig. 1 The structure of a decision tree

$$f(x) = y + \sum_{i=1}^{n} feature.contribution(i, x)$$

where x is an individual instance, $f(x)$ is the prediction of an individual instance, y is the mean of the target outcome, and n is the total number of features. The prediction of an individual instance is the mean of the target outcome plus the sum of feature contributions of n features.

Random Forest (RF) [19] is a supervised ML algorithm that operates by generating a multilevel DT. RF is widely used for classification and regression. The RF algorithm comprises two main parts: bagging and boosting. Bagging refers to the creation of additional data by replicating original data to reduce the variance. Boosting refers to the sequential combining of weak learners with strong learners. The formula for RF can be expressed as follows:

$$y = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{n} W_j(x_i, x')y_i$$

where m represents the number of trees, y represents the output, x' represents the new point wanted to be predicted, W denotes the weight value, j represents the respective tree, and x_i denotes the neighbor of x' that share the same leaf in tree j.

Extra Trees [5] is a supervised ML algorithm that shares similarities with RF. The main difference is that RF uses bootstrapping that sub-samples input data with replacement, whereas Extra Trees uses all original samples. During the splitting-node phase, RF chooses the optimal split, whereas Extra Trees chooses the split randomly. The extra trees algorithm is faster but does not return the optimal tree.

Naive Bayes [100] is a supervised ML algorithm based on the Bayes theorem, which states that given a class of variables, every feature is conditionally independent. The Naive Bayes function can be expressed as follows:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(xi|y)$$

where x represents the features, n denotes the total number of the features, and y represents the class variable.

Gradient Boosting (GB) [35] is an ensemble learning method for modifying weak learners by unifying them into one stronger learner. The GB method is widely used with DT as the learning model. The residual learning and decision path of GB trees can be used to measure the contribution of each feature to the prediction result, making the GB model is explainable. The GB formula for DT can be expressed as follows:

$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} 1(x \in R_{jm})$$

where F_m represents the model, n_j denotes the number of samples in terminal node j, and $\gamma_{jm}1(x \in R_{jm})$ implies that the value of Y_{jm} is chosen if the given x is located in the terminal node R_{jm}. The average residual r_{im} in the terminal node R_{jm} is the optimal value of Y_{jm} that minimizes the loss function.

3.3 XAI methods

Explanation methods are used to identify the contribution of each data parameter to the classification made by ML algorithms. In this section, we introduce such methods by classifying them into two families: local explanation and global explanation methods.

3.3.1 Local explanations

Local explanations are used to understand the classification of a single data input. For example, "Why is this image classified as a cat?"

Local surrogate models is a model that accurately do approximation in a local feature space around a single input, explaining an individual prediction. A surrogate model itself is a statistical model that has been trained to accurately approximate the output of a black-box model. One example of local surrogate model for XAI is Local Interpretable Model-agnostic Explanations (LIME) [97]. LIME is an explanation method that locally approximate a black-box ML model to explain each prediction. The main idea of this model is to perturb the original data, and then feed them to the model. The data points are weighed as the proximity function of the original point. Based on those data points, LIME trains a local surrogate model that locally gives a good explanation. The local surrogate model can be described by the following formula:

$$explanation(x) = argminL(f, g, \pi_x) + \Omega(g)$$

where x represents the instance for generated local model g that minimizes the loss function L, f denotes the original model, $\Omega(g)$ represents the model complexity, and πx denotes the proximity measure that defines the area around instance x considered for the explanation.

LIME was extended using a Bayesian approach in [116, 140] because of its instability. If we perform a repeated run using LIME, it will generate inconsistent explanations. To address this issue, studies have used Bayesian reasoning to exploit the prior knowledge and improve the explanation fidelity.

SHapley Additive exPlanations (SHAP) [78] is a method to explain individual predictions in a black-box setting. The prediction is based on the Shapley value—an average contribution value of a feature across all possible combinations. The main purpose of SHAP is to measure the contribution of each feature to the prediction result. It can be described using the following formula:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

where g denotes the explanation model, z' denotes the combination vector, M denotes the maximum combination size, and ϕ_j denotes the feature attribution for feature j. Feature attribution indicates the contribution level of each feature to the prediction result. SHAP was extended with a Bayesian approach in [116] because of its high computational cost. In addition, a prediction algorithm that approximates the optimal number of samples is required to reduce the number of queries.

Anchors modeling [98] is an explanation method on any black-box model that attempts to find a decision rule for the prediction process. A rule becomes an anchor of a prediction if changes in any feature value will not affect the prediction result. Anchors combine graph search and reinforcement learning methods to minimize the processing time. Anchors use perturbation-based methods to generate local explanations using if–then rules. This differs from LIME, which uses a local surrogate model. Anchor modeling is a model-agnostic method that can be applied to any model. An anchor A can be expressed as follows:

$$\Xi_{Dx(z|A)}[1_{f(x)=f(z)}] \geq \tau, A(x) = 1$$

where Ξ represents the evaluation function, x represents the explained instance, f denotes the classification model, $Dx(z|A)$ represents the distribution of neighbors x—where the same anchor A is applicable—and τ specifies the precision threshold, which is between 0 and 1.

Individual conditional expectation (ICE) [41] is an explanation method that uses a line plot for each instance to demonstrate the degree of variation in predictions when a feature is modified. ICE focuses on a specific instance and visualizes the prediction dependence of each feature separately. Thus, it can uncover a heterogeneous relationship with an intuitive curve that is legible. However, ICE can only display one feature at a time. There can also be some invalid data points if the feature of interest correlates with another feature.

Counterfactual explanations [130] represent a causal scenario that can be described as "If A does not happen, B will

not happen." When applied in XAI, this concept describes the smallest change in feature values that can affect the output. Counterfactual explanations can be applied to both model-agnostic and model-specific scenarios. A counterfactual instance must generate predefined predictions as close as possible with similar instances regarding feature values. A counterfactual explanation can be formulated as follows:

$$arg \min_{x'} \max_{\lambda} L(x, x', y', \lambda)$$

where x denotes the selected instance that is to be explained, y' represents the desired outcome, λ represents the bias value, and x' represents the counterfactual explanations. A low initial value is assigned for λ, which is then continuously increased until the loss is minimized. Finally, the list of counterfactual explanations that minimized the loss is obtained.

Local explanation method using nonlinear approximation (LEMNA) [45] is an XAI method for AI-based security applications. It combines a mixture regression model with a fused lasso to generate high-fidelity explanation results. The fused lasso is used to handle the feature dependency problem. The mixture regression model is used to approximate local nonlinear decision boundary explanations for complex security applications. The formula for LEMNA can be described as follows:

$$f(x) = \sum_{j=1}^{K} \pi_j(\beta_j.x + \epsilon_j)$$

where K specifies the number of linear models, ϵ denotes random variables from a normal distribution, β denotes the regression coefficient, and π holds the weight value.

3.3.2 Global explanations

Global explanations are the opposite of local explanations in that they are focused on the overall behavior of the model. Instead of explaining singular instances, they target the average distribution of data.

Partial dependence plot (PDP) [35] is an explanation method that illustrates the marginal effect of a feature on the output of an AI model. The PDP focuses on the overall average instance, instead of a specific one. Thus, it is also the opposite of ICE. The PDP can be considered the average line of an ICE plot. The value for one instance can be computed by setting all other features with similar values, and then creating another variant for that specific instance. As such, the PDP can reveal the relationship between a feature and the prediction result. The formula for a PDP can be described as follows:

$$\hat{f}_S(x_S) = E_{X_C}\left[\hat{f}(x_S, X_C)\right] = \int \hat{f}(x_S, X_C) d\mathbb{P}(X_C)$$

where $\hat{f}$ denotes the partial function, x_s represents the input feature that is going to be plotted, and x_C consists of other features used in the model.

Accumulated local effects (ALE) [36] explains the influence of a feature on the prediction result of an AI model on average. The concept of ALE was introduced to address the main limitation of a PDP: its fidelity level reduces drastically if the features in the AI model are correlated. ALE show the variation of model prediction in a small area where the analyzed input is located.

Global surrogate model [90] is an explainable method for generating a surrogate model by approximating the prediction result and the interpretability of the underlying explainability model. First, a dataset is selected (it can be the same dataset that was used to train the underlying model or a new dataset). Then, for the selected dataset, the prediction result is derived from the original model. Subsequently, an interpretable model is trained based on the dataset and its prediction. Finally, a global surrogate model is generated.

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(y_s^{(i)} - y_o^{(i)})^2}{\sum_{i=1}^{n}(y_o^{(i)} - y_{mean})^2}$$

where R^2 is the coefficient of determination that represents the proportion of variation, SSE is Sum of Squares Error, SST is Sum of Squares Total, $y_s^{(i)}$ represents the i-th instance of the surrogate model, $y_o^{(i)}$ represents the prediction result of the original black-box model, and y_{mean} represents the mean of the original black-box model prediction.

Feature interaction [137] is an explainable method based on marginal distribution estimations. It was proposed to address the problem that when features are correlated, the prediction cannot be manifested as the sum of feature effects. The effect of one feature influences other features. The feature interaction concept states that the interaction between features represents a change in the prediction result, which happens by varying the features considering each feature effect. The formula of feature interaction can be expressed as follows:

$$PD_{jk}(x_j, x_k) = PD_j(x_j) + PD_k(x_k)$$

where $PD_{jk}(x_j,x_k)$ represents the partial dependence function between features and $PD_j(x_j)$ and $PD_k(x_k$ are the partial dependence functions of each single feature.

Functional decomposition [30] is a method that constructs a visualization of individual features and interaction effects. The prediction function can be represented as the sum of

functional components. The decomposition function can be described as follows:

$$f(x) = \sum_{s \subseteq \{1,\ldots,p\}} f_s(x_s)$$

where p denotes the number of input features, f denotes the prediction function, and x_s denotes the feature vector in the index set S, in which each subset represents a functional component.

3.3.3 Explaining neural networks

The methods presented here are specific to gradient-based models, which use forward and backward propagations of gradients during training.

Feature visualization [92] is an explainable method that converts learned features into an image visualization that describes the feature's characteristics. DNNs can learn high-level features through hidden layers. An input image undergoes several feature engineering processes as it passes every layer. The deeper the layer, the more complex the learned features become. The feature visualization of a layer in a neural network (NN) is done by finding the input that maximizes the activation of that layer. The optimization problem of feature visualization can be described as follows:

$$img^* = \arg\max_{img} \sum_{x,y} h_{n,x,y,z}(img)$$

where h denotes the activation function, image img is the input of the neural network, b describes the layer, x and y specify the spatial positions of a neuron, and z is the channel index.

Saliency maps [9]—also known as pixel attribution, heat map, sensitivity map, gradient-based attribution method, or feature relevance—are an explanation method that explains individual predictions by providing attributes to each feature based on its degree of influence on the prediction result. Saliency maps can be classified into two approaches: perturbation- and gradient-based approaches. The perturbation-based method generates an explanation by manipulating parts of an image, which is categorized as model-agnostic. The gradient-based approach computes the gradient of the prediction result with respect to the input features. Both approaches assign each pixel a value that can be converted into the categorization result's relevance level.

Explanatory graph [138] is a graphical model that represents knowledge in each convolution layer of a convolutional neural network (CNN). As the filter in a pretrained CNN is activated by different object parts, the patterns from each filter are extracted. Subsequently, the patterns are disentangled in a supervised manner to generate an explanatory graph that clarifies the knowledge representation. The method visualizes the spatial relationships between patterns, filtering out noisy patterns, and ensuring the consistency of feature representation among different images.

4 Explainable classification for cybersecurity

In this section, we survey studies in three different scenarios: when explanations are used for transparency of the model; when explanations are used to improve the performance of the model; and when explanations are used to explain errors made by the classifier. We summarized the contents of these studies in Table 2.

4.1 Explanations for transparency and trust

A core objective of XAI is to provide users with actionable explanations that will help them understand why the model made a decision. We divide this section by the type of explanation: surrogate models, global explanations, and interpretable models.

4.1.1 Explaining with surrogate models

Islam et al. [54] proposed a semantic approach assigning confidentiality, integrity, and availability (C, I, A) meanings to attacks and features in a dataset. First, they designed a feature generator, where the three most important features for each attack are extracted. New features were then created by assigning a C, I, or A attribute to the ones they extracted. In addition, they computed a coefficient that illustrated the weight of the feature in the overall impact of C, I, or A. Second, they implemented an evaluator that ran multiple attack detection models and measured the impacts of different sets of features, including the previously generated ones. They used the CIC-IDS2017 dataset [110]. Due to resource limitations, the dataset could not be entirely exploited, thereby limiting the scope of validity of their results. The results illustrated that, although the set of generated features did not outperform the full set of features, it did not perform significantly worse. The generated features also provided a human operator with an explanation of the contribution of the C, I, or A attribute in the classification of the attack. Similar results were obtained when trying to detect unknown attacks (i.e., removing all data from a specific attack before training). Interestingly, a structured query language injection can only be detected using the proposed method. The weaker performance is partly due to the simplification of the attacks and features into three attributes (i.e., C, I, and A).

Table 2 Comparison of studies using XAI for cybersecurity tasks

	Linear/Logistic Regression	Decision Tree	Random Forest	Extra Trees	SVM	Naive Bayes	Gradient Boosting	Multilayer perceptron	ANN	Deep Feed Forward	Autoencoder	CNN	Other	SHAP	LIME	SHAP/LIME variations	LEMNA	Decision Tree	Adversarial Examples	Saliency maps	Feature Visualization	CAM (+variations)	Other	Dataset
Explanations for transparency and trust																								
[54]		✓		✓		✓																	✓	CI-CIDS2017
[143]									✓														✓	WUSTL-IIoT, NSL-KDD, UNSW
[57]		✓					✓		✓					✓	✓									Private dataset
[123]									✓								✓							CIC-IDS2017
[45]													✓											MINIST + PDF Malware
[132]					✓									✓		✓								NSL-KDD
[11]											✓			✓		✓								KDD99
[106]		✓								✓				✓										CSE-CIC-IDS2018, NF-CSE-CIC-IDS2018-v2
[8]			✓				✓									✓								ISCX-URL2016, CICMalDroid 2020
[58]													✓		✓									ToN IoT dataset
[124]		✓												✓										CIRA-CIC-DoHBrw-2020
[39]	✓															✓								Clusit cyber loss database
[53]												✓									✓			Malware binary dataset
[109]													✓							✓				National Vulnerability Database
[79]		✓												✓										KDD99
[133]		✓															✓							Network traffic
Explanations for improving the performances of the classifier																								
[131]			✓											✓										CSE-CIC-IDS2018
[43]			✓									✓			✓					✓				Dataset collected from FireFox and Chrome
[102]										✓				✓										CIC-IDS2017
Explaining errors																								
[82]	✓								✓										✓					NSL-KDD
[32]													✓	✓										AndroZoo
[76]	✓												✓									✓		UGR'16
[34]	✓	✓	✓										✓									✓		Private dataset

Zolanvari et al. [143] proposed TRUSTXAI, a model-agnostic, high-performance XAI model suitable for numerical applications. They used three different datasets: WUSTL-IIoT [142], NSL-KDD [126], and UNSW [88]. These are tabular traffic datasets for intrusion detection. The system works by modeling the statistical behavior of AI outputs. The input features are transformed into a set of latent variables using a factor analysis [93]. Subsequently, these variables are measured using the mutual information concept. The most influential variables for the output are set as representatives of the class. Finally, the Gaussian distribution is used to determine the likelihood of each sample's class. Their experiment showed that TRUSTXAI successfully provided explanations for random samples with a 98% success rate. Six samples (three positives and three negatives) were randomly chosen from the test set. Compared with LIME, TRUSTXAI was superior in performance, speed, and explainability. They claimed that the proposed model yields a more straightforward explanation than LIME, by modeling the output with a statistical measure, which is easy to understand. However, it has one main limitation; it tends to overfit due to the use of mutual information when picking class representatives.

Karn et al. [57] introduced an automated cryptomining pod detection in a Kubernetes cluster using a statistical explainability mechanism. They attempted to identify and classify any background malware executables that were running. For the explanation task, they implemented SHAP for XGboost, LIME for NN, and DT. The explanation justifies any pod removal decision, implying a running process of cryptomining. Similarly, [123] proposed a hybrid oracle–explainer approach to develop an explainable intrusion detection system (IDS), which combined an opaque classifier based on an artificial neural network (ANN) and an interpretable module using DT. During the inference phase, the ANN classification decision of a given input is explained by the DT's decision on the same input, or the closest input in terms of the l^2 (Euclidean) norm.

Guo et al. [45] proposed LEMNA, an explanation framework for malware detection applications. LEMNA is the sole explanation method designed specifically for cybersecurity applications. They used a the PDF malware dataset from Smutz et al. [120]. LEMNA attempts to approximate a local area within a DL decision boundary using an interpretable model. This model is designed to handle feature dependency and nonlinear local boundaries. The framework works by treating a DL model as a black-box and performing approximation using a mixture regression model boosted by a fused lasso. The fused lasso forces similar coefficients of neurons to be assigned to adjacent features within a small threshold, taking features as groups and making the learning algorithm learn a target model based on the feature groups. The mixture regression model is a combination of multiple LR models; Guo et al. employed it to avoid the nonlinear approximation problem.

4.1.2 Global explanations

An explainable IDS was proposed in [132], which combined local explanation (using original SHAP) and global explanation (using modified SHAP). In particular, the value of each feature was divided into several intervals, each of which was measured with Shapley values. The Shapley values were then averaged, yielding a global explanation. While SHAP offers fast computation for explanation, the framework lacks capability for real-time updates.

In [11], an autoencoder (AE)-based anomaly detection scheme using SHAP is proposed. They examined the robustness of the methodology by replacing one feature with noise and assuming that the noise feature should not explain an anomaly. If the selected feature contributed to the explanation, they introduced a perturbation. The new instance should be less anomalous, and the anomaly score should then be reduced. They experimented with the KDD Cup 1999 dataset (intrusion detection) from the UCI ML archive, revealing that SHAP outperformed LIME on reducing the reconstruction error.

SHAP was also used in [106] for explaining and interpreting classification decisions of an ML-based network IDS. Two classifiers—a deep feedforward NN and an RF—were evaluated on several recent intrusion detection databases, namely CIC-IDS2018 [22], TON_IoT [87], and BoT–IoT [63]. Two feature sets of each database were considered: one set contained 83 features extracted via CICFlowMeter [71], while the other contained 43 features extracted from NetFlow [107]. The evaluations were focused on finding the most interesting features for each classifier. The results of *explainability* exhibited some similarities between the two classifiers with respect to the most influential features in different databases. The authors also noted that the influence and importance of each network feature varied with the dataset.

Alenezi et al. [8] designed an explainable ML framework for malware and malicious URL detection. They implemented three XAI methods based on SHAP: TreeExplainer, KernalExplainer, and DeepExplainer. These XAI methods explained the prediction of common classifiers, such as RF and XGboost. They used a URL dataset, ISCX-URL2016 [81], and an Android malware dataset, CICMalDroid 2020 [80], both published by the Canadian Institute for Cybersecurity. They compared the performance of each method using different setups. The results showed no optimal universal setup for any one scenario.

Khan et al. [58] designed a timely detector for attack vectors on the Internet of Medical Things networks. The model was developed using bidirectional simple recurrent units.

The detector uses the phenomenon of skip connections to solve the vanishing gradient problem and reduce the training time. The study used the TON_IoT dataset [87], which contains several cyberattack classes, such as ransomware, backdoor, denial-of-service, distributed denial-of-service, Man-in-the-Middle, injection, cross-site scripting (XSS), and scanning attack. Their experiment demonstrated that the proposed model outperformed the long short-term memory and gated recurrent unit models at a lower computational cost. They also used LIME to investigate the contribution of each feature in the prediction phase.

Zebin et al. proposed an RF-based IDS to detect DNS-over-HTTPS (DoH) attacks [124]. Precisely, they used SHAP to explain the results by determining the most important features. The proposed method was evaluated with the CIRA-CIC-DoHBrw-2020 [86] dataset, where the traffic is described by 29 features. The experimental results demonstrated the *flow duration* and *packet length* as the most discriminative features.

Similarly, Giudici et al. [39] applied an enhanced version of SHAP (which is more global and robust in the presence of outlying observations), called the Shapley–Lorenz decomposition method [40], to explain classification decisions on ordinal cyber-data (e.g., ordinal severity levels of cyber-risks include "low," "medium," or "high"). To explain the severity of each event, a linear rank regression model was used to express the observed severity as a function of a set of four explainable variables describing i) attack type (e.g., cybercrime or espionage), ii) attack technique (e.g., zero-day or malware), iii) type of victim (e.g., banking or hospital), and iv) geographic area where the event occurred (continent).

The explainable variables were the marginal contributions associated with each feature and are calculated using the Shapley–Lorenz decomposition method. The results showed that Shapley—Lorenz values were significantly easier to be interpreted than the Shapley values that were not normalized.

Iadarola et al. [53] introduced a framework for Android malware detection using an explanation method for image recognition. They tried to address the weakness of signature-based anti-malware detection, which cannot detect zero-day malware. First, a binary executable dataset was converted into images, and then a CNN was used for training. In the explainability steps, they generated heatmaps with Grad-CAM [108] and classified a subset of the test samples to the corresponding class of malware. They evaluated 8,446 Android malware samples from 6 malware families and obtained an average accuracy of 0.97. In addition to its inability to detect unidentified malware, the method requires a huge amount of training data to achieve a decent detection accuracy.

Shahid et al. [109] proposed an automated common vulnerability scoring system (CVSS) vector and severity score calculator for security vulnerability detection. The CVSS standard is an analysis of the severity of computer vulnerabilities conducted by security experts. The CVSS vector represents the characteristics of a vulnerability, which can be computed into the severity score. The severity score represents the level of danger posed by the vulnerability and acts as the threshold in the classifier. CVSS scores are usually designed by a human expert, which is a time-consuming and arduous process. Consequently, automation is required. They trained several bidirectional encoder representations from transformer classifiers, with each metric producing the CVSS vector. The goal of the trained model was to determine the value of a CVSS vector with high accuracy. For the explainability method, gradient-based input using a saliency map was used to determine the most influential input.

4.1.3 Interpretable models

Mahbooba et al. [79] aimed at addressing the trust issue between users and ML models for IDS. They highlighted that most previous studies focused on the accuracy of classifiers without providing any insight into their reasoning or behavior. They used the DT algorithm on the KDD99 dataset [126]. In summary, their methodology comprised three main steps: feature ranking, DT rule extraction, and comparison with state-of-the-art algorithms. They described the feature on each branch and the threshold value to explain how the tree made decisions. However, their algorithm was vulnerable to overfitting amid noise in the dataset.

Wang et al. [133] introduced TrafficAV, an explainable mobile malware detector. They captured the network traffic of a mobile device and investigated it for suspicious activity. TrafficAV gathers network traffic features by performing a multilevel network traffic analysis using the C4.5 DT algorithm. The explainability method is ad hoc, based on the DT. They evaluated 8,312 benign applications and 5,560 malware samples on HTTP models, achieving 99.65% accuracy.

4.2 Explanations for improving the performance of classifiers

In this subsection, we discuss recent works about using XAI to improve the performance of classifiers. The majority of latest publications in the field of AI have focused on improving accuracy, detection rate, and F_1 score, while reducing the false alarm rate. Improving the model's performance via parameter tuning in a heuristic manner is computationally taxing. Consequently, XAI is deemed capable of increasing AI's performance in an explainable way. We highlight three use cases: 1. Explainable IDS, 2. Side channel attack detection, and 3. Anomaly detection.

Khan et al. [131] leveraged global explanation on TreeSHAP to correlate an RF decision for explainable IDS. Their explainable IDS architecture comprised three main

modules: (1) RF classifier (RFC) module for security predictions; (2) SHAP module extracting values relative to each feature of the dataset, and representing the importance of each feature in the decision made by the RFC; (3) Credibility assessment module (CAM), which utilizes the prediction and Shapley values to evaluate the confidence of prediction made by the RFC.

In particular, the CAM module evaluates the plausibility of the second-most probable prediction over the confidence expressed by the Shapley values of the most probable prediction computed by the RFC module. In the case of divergence, other classifiers were used to reassess the decision. The CIC-IDS2018 dataset [22] was finally used to compare the proposed IDS with other state-of-the-art classifiers. Adversarial attacks were also employed to evaluate the robustness of the proposed IDS, which outperformed a native version of the RFC.

Gulmezoglu [43] proposed an XAI-based framework using side-channel analysis against website fingerprinting attacks. The framework detects side-channel attacks by discovering the most dominant features extracted from a dataset using CNN and RF. During the training phase, they used a self-generated side-channel attack dataset collected from the Google Chrome and Firefox browser developer tools. After the DL model was trained, LIME and saliency maps were used to examine the most dominant features of the website fingerprinting attack. They also verified the robustness of the framework. After perturbing the data points, a new model was trained and tested with the perturbed dataset. Their experiment revealed a drop of 16% in the attack rate, with five times less performance overhead. Subsequently, they generated adversarial noise to anticipate further attacks.

Roshan et al. [102] proposed the application of KernelSHAP to reconstruct the errors of an AE to select the best features in an anomaly detection dataset. To understand and interpret the roles of features in improving AE-based anomaly detection, they compared three cases: (1)] no feature selection (all features); (2) feature selection using unsupervised feature correlation; (3) feature selection using SHAP.

Experiments with a subset of CIC-IDS2017 (benign data for training, benign and malicious data for testing) revealed SHAP-led feature selection as exhibiting the best overall performance. However, it displayed slow feature computation and the possibility of an increase in the time complexity with the sample size.

4.3 Explaining errors

A few studies have explained, using XAI, why a security model would make a mistake. Marino et al. [82] used adversarial examples to explain the importance of each feature on the decision made by gradient-based classifiers. They examined a set of misclassified samples and found the minimum amount of modification required to rectify the classification. They compared two models, an LR and a multilayer perceptron [101]. They used the NSL-KDD dataset [126]. However, the dataset was published in 2009. This limitation does not influence the usability of the proposed method; however, it does challenge the validity of the results for current threats. Although the study allowed determining the key features in the model, it did not provide interpretation guidelines about the meanings of the features.

Fan et al. [32] used SHAP for feature attribution computation in an Android malware detection system. The main purpose of their study was to maintain model classification performance over time. They stated that the change in performance was difficult to understand when they updated the model. SHAP was used to interpret the output of the model by assigning Shapley values for each feature. The prediction change was analyzed by comparing the pattern changes of feature attribution. First, the feature attribution of each sample before and after the update was collected. Second, the changes in feature attributions were clustered, obtaining the pattern of changes. The experiment showed that the method successfully prevented overfitting and ineffective updates.

Liu et al. [76] introduced FAIXID, a framework that provides data cleaning and XAI for IDS. Unlike other studies, they offered a data cleaning method to address the data quality problem. The cleaning process solved the issue of "the data we want and the data we have." Their focus was to propose several data cleaning techniques, instead of the explainability method. However, their method was designed specifically for a homogenous set of features—if the dataset has different features, more adjustments will be needed.

Farrugia et al. [34] proposed the usage of XAI in an application for cyber-fraud detection, with the goal of achieving a fully autonomous prescriptive solution for explainable cyber-fraud detection within the iGaming industry. The application of XAI in this context allowed the authors to minimize the adverse effects of incorrect predictions. A private dataset with labeled instances of verified fraudulent players made by the Gaming Innovation Group was used. However, this manually labeled dataset was also the main limitation. In their method, they trained different models (RF, LGB, DT, and LR) on the dataset; they used stratified 10-fold cross-validation and compared the models using the area under the curve (AUC). Next, they extracted explanations for every individual prediction. Finally, they empirically evaluated data drift and suggested retraining the model every month as the drift rate was approximately one month. They compared each model using the standard metrics (AUC, precision, time, recall, and F1), where RF obtained the most consistent results.

4.4 Discussion

Despite not belonging to a cybersecurity use case, the merits of the following study deserves the reader's attention. Aguilar et al. [4] proposed an explainable AE using DTs for anomaly detection that deals with categorical data. Particularly, categorical attributes were one-hot encoded, and each attribute was used to construct a DT, which output the predicted attribute by considering other attributes. A final prediction layer was employed to determine whether the data point was an anomaly. As such, each DT explained one attribute, explaining the final decision through a set of DT rules. Twenty-eight datasets from the UCI ML Repository [14] were used for the experiments. Especially, datasets with two balanced classes or multiple imbalanced classes were divided using distribution optimally balanced stratified cross-validation [136]. The resulting datasets had one majority class and one minority class for anomaly detection. The performance of the AE was compared with six other classifiers in terms of AUC and average precision score. The proposed solution ranked third per median value for AUC, but first for average precision. The strength of their study is the simplicity with which explanations were extracted from the trees while demonstrating them through synthetic examples. However, this system had limited scalability to datasets with less than a thousand attributes or with categorical attributes having tens of different values. As listed in Table 2, DTs have already been used in cybersecurity tasks, and the performance of this DT-based AE approach is worth investigating.

5 Security properties of XAI

In this section, we survey the literature investigating the security of XAI and outline several relevant security properties. We characterize the security of explanation methods according to four properties: fairness, integrity, privacy, and robustness. We include these properties under the scope of cybersecurity by considering how an attacker may compromise them (e.g., altering explanations for an unfair treatment) or how a potential victim may defend against such attacks (e.g., by consolidating explanations against adversarial examples). By analyzing the literature, we elaborate on each property in the following subsections. The surveyed works are summarized in Table 3.

5.1 Fairness

A model is said to be *fair* if its output is irrelevant to individuals' sensitive features, such as sex, race, or religion [95]. This is especially crucial when models used for decision-making affect individuals, such as loan, employment, insurance, or sentence. An explanation can be used to evaluate the fairness of a model, e.g., an auditor can verify the fairness of a decision by inspecting how a local model produced by LIME or SHAP uses the sensitive features. However, an agent who plans to discriminate or favor a group of people would attempt to deceive the auditor into believing the model is fair. Such an attack, called *fairwashing*, has a notably different threat model than classic attacks on ML models, because a fairwasher is generally the model owner. The threat model is advantageous to the attacker because he/she generally has full knowledge of and control over the model, whereas the auditor may only possess limited knowledge and no control. Fairwashing has been widely studied, and most explanation methods thus far have been successfully attacked. The following studies show that, in general, an owner can easily fairwash their model and complicate the detection of fairwashing for the auditor. These studies could serve as a reminder that the model owner should not be allowed to generate explanations.

Anders et al. [10] showed through differential geometry that the saliency-based explanations of a classifier can be arbitrarily modified without changing its predictions. They demonstrated the relationship between explainability and manifold learning, highlighting that the explanations were based on dimensions orthogonal to the data manifold. They also proposed and experimented with a robust explanation method called *tangent-space-projected explanation*, which could not be manipulated by the attack described in the study.

Slack et al. [117] focused on manipulating counterfactual explanations. These explanations are especially attractive because they are actionable, i.e., they indicate what modification of the input can modify the output. This explanation is generated by a local search that attempts to maximize an objective function. This objective function depends on the endpoint loss (how close its value is to the desired class) and the distance between the origin and the endpoint. For NNs, this explanation is generally computed by hill climbing. A fair model should provide explanations with actionability (i.e., the simplicity to act on the explanation) that does not depend on the sensitive features, such as sex or race. However, the authors showed that the model owner can modify the learning procedure, so that a small perturbation applied to an input belonging to some population leads to counterfactual explanations. Precisely, the Euclidean norm of the counterfactual explanation will be remarkably lower. The authors experimentally demonstrated that such a modified model retained good accuracy, and the maneuver was not easily detected. Countermeasures include reducing the model complexity and adding noise to the initialization of the counterfactual-explanation-generation procedure.

Table 3 Comparison of works targeting the security of XAI

	Fairness	Integrity	Privacy	Confidentiality	Robustness		Attack	Countermeasures		Deep Neural Networks	Autoencoder	CNN	Other		SHAP	LIME	SHAP/LIME variations	LEMNA	Gradient Based	CAM based	Saliency maps	Feature importance	Feature Visualization	Other	Dataset
[116]	✓						✓			✓														✓	Communities and Crime, German Credit
[49]		✓					✓					✓							✓	✓	✓				ImageNet
[10]	✓						✓			✓		✓							✓		✓				MNIST, FashionMNIST, CIFAR10, German Credit
[115]		✓					✓						✓			✓			✓		✓				SST-2, AG News, IMDB
[28]		✓					✓					✓							✓		✓				ImageNet, CIFAR-10
[117]	✓						✓						✓		✓	✓									COMPAS, Communities and Crime, German credit
[72]	✓						✓			✓														✓	German credit
[27]	✓						✓			✓						✓									Adult, Compas, German, Bank
[7]	✓						✓						✓											✓	Adult Income and ProPublica Recidivism
[37]		✓					✓					✓								✓					Dog vs. Cat dataset, the UIUC sports event dataset
[66]	✓	✓	✓	✓	✓		✓			✓	✓								✓						Malicious PDF, Malware, IDS
[65]			✓	✓			✓			✓			✓												Passwords, CIC-IDS2017, Malware
[134]					✓			✓		✓															CIFAR-10,Flower
[20]		✓						✓				✓												✓	MNIST, CIFAR10
[139]		✓					✓			✓															ImageNet, ISIC 2018 challenge
[141]			✓				✓					✓							✓	✓					iCV-MEFED face, CelebA,MNIST
[113]			✓				✓						✓												Purchase and Texas datasets, CIFAR-10 and CIFAR-100 Adult dataset, Hospital dataset (Strack et al. 2014)
[84]				✓			✓					✓	✓												CIFAR10, SVHN
[56]					✓		✓			✓															ImageNet
[112]					✓		✓					✓												✓	ImageNet
[38]					✓		✓			✓															ILSVRC2012 CIFAR-10
[99]	✓							✓				✓													ImageNet
[69]					✓		✓					✓								✓				✓	ImageNet
[59]					✓			✓				✓									✓				MNIST
[70]					✓			✓		✓		✓												✓	sentiment analysis datasets (SST,Twitter,IMDB)
[83]					✓		✓												✓						Drebin, Androzoo
[17]					✓			✓				✓													MNIST, CIFAR10, ImageNet

Slack et al. [118] indicated that ad hoc explanations such as LIME and SHAP, which work by evaluating the neighbors of the input for which they generate an explanation, can be deceived. However, the neighbors may be outside the manifold, i.e., they have a very low probability in the distribution of the input feature space. The authors exploited this fact to build an off-manifold point detector that detected whether LIME or SHAP were using some neighbors to generate an explanation. When this explanation process is detected, the model owner can substitute their model with a locally fair model that only relies on relevant attributes. The authors experimentally revealed that LIME was more vulnerable than SHAP. However, this attack can only be performed in black-box settings, when the auditor does not have access to the model's internals. Additionally, its quality greatly depends on the off-manifold point detector, which must have good accuracy.

Sinha et al. [115] introduced an attack that completely modified the explanations with subtle input perturbations, while leaving the prediction unchanged in the domain of NLP applications. They built the input perturbation using a greedy algorithm based on metrics adapted to NLP applications, and successfully deceived LIME and integrated gradient explanation techniques.

Le Merrer et al. [72] indicated how a model owner can always wipe unwanted attributes from a local explanation when he/she controls the explanation generation method and not an external auditor. Precisely, they focused on local explanations in the form of DTs, where a model owner could simply eliminate any occurrence of sensitive features in the DT after its generation (e.g., with LIME). This attack could be detected by exhibiting inconsistencies between explanations; however, the authors showed that these inconsistencies were difficult to obtain in practice, at least when the search was random.

Diamnov et al. [27] proposed a method that could modify a pretrained model to manipulate the output of feature importance explanation methods. They assumed that the explanation methods used in real-world settings could not indicate the fairness of a model. To prove their hypothesis, they chose several datasets from the UCI ML Repository [14], which contains data with sensitive features, such as sex or race. They optimized an existing model with a modified loss objective function. After the attack, the feature importance computed by the explanation method was completely modified.

Aivodji et al. [7] investigated the rationalization problem and the associated risk of fairwashing. Subsequently, they introduced LaundryML, an algorithm that enumerated the optimal model according to fidelity and unfairness. They considered fairwashing when the fidelity of the new model is high while the unfairness is significantly lower. They used two datasets known for their biased nature—Adult Income [62] and ProPublica Recidivism [55]. They used the fidelity and unfairness metrics to evaluate the performance of LaundryML. The results obtained using a real-world dataset demonstrated the feasibility of the proposed approach.

Hence, all those attacks on fairness are fairwashing, where the model owner attempts to deceive auditors into believing their model was fair. However, fairwashing is not the only type of attack that can be orchestrated on fairness. For example, a user could create an adversarial example to manipulate a fair model into producing explanations that make it look unfair. If technically possible, this attack could be used to tarnish the public relations of a company.

Relevant use cases of fairwashing applied to security applications have not been published. Most scholars are interested in social bias, particularly linked to sex and skin color. Nevertheless, we expect fairwashing to be applicable to nationalities or geographical locations at a time where security products are playing an increasingly crucial role in cyberwarfare. Although fairwashing is widely studied, few defense mechanisms exist. In fact, several proposed attacks have resulted in their theoretical concealability. This line of research should be expanded to achieve verifiable fairness of ML models.

5.2 Integrity

Integrity is a classic property of data and processes that concerns their trustworthiness and accuracy. A classic attack on the integrity of an ML model is the set of adversarial examples that evade a classifier by subtly modifying the feature input. Fairwashing can be considered a subcategory of integrity attacks, as it targets the integrity of explanations to feign fairness. In this subsection, we focus on integrity attacks that do not target fairness.

An explanation typically can be used as a second layer of information to complete a prediction. For example, an expert could use both prediction and explanation to make a decision. Therefore, an attack on integrity, in the classic sense, is an attack altering data provided to the expert (prediction, explanation) with no countermeasures indicating the manipulation. Notably, during fairwashing, the objective of the attacker is to retain the (biased) output and modify the explanation; meanwhile, in the class of attacks we describe in this subsection, both the output and explanation are manipulated.

The method proposed by [49] comprised optimizing a model to deceive saliency map-based explanation methods. This optimization is achieved by modifying the loss function to include the error between the produced explanation and the targeted, artificial explanation. The attack can be either passive, in which case the produced explanations are uninformative to the auditor, or active, in which the model owner fabricates the explanations. The authors experimentally demonstrated the transferability of their attack to various saliency map-based explanation methods.

Dombrovski et al. [28] showed that subtle modifications in images processed by NNs could yield arbitrary explanations for saliency map methods, while keeping the network output constant. They orchestrated the attack by local optimization, similar to classic adversarial attacks. The authors demonstrated that they could achieve this primarily owing to the use of the ReLU activation function. Indeed, the attacks were not distinguishable, creating piecewise linear boundaries and a very large curvature. The authors proposed more robust explanations by replacing ReLU with SoftPlus only in the network that generated the explanations.

Galli et al. [37] analyzed the impact of adversary attacks on XAI methods. They used two perturbation attacks (IFGSM, DeepFool) on four CNNs to generate adversarial samples. Their experiments were performed on the Dog vs. Cat dataset [31] and the UIUC sports event dataset (Event8) [73]. Two methods of XAI (layered Grad-CAM and guided Grad-CAM) were used to detect adversarial attacks. However, the authors showed that the attack had no discernible impact on the interpretations produced by XAI to someone who is unaware of the attack, as the change was not detectable. Thus, relying on the explainability of results to detect the existence of adversarial attacks is not a rigorous approach.

Kuppa et al. [66] proposed a black-box attack on explainable models and evaluated the attack on three datasets. The attack had two possible targets: either the classifier alone or the classifier and interpreter. The attack focused on gradient-based classifiers (i.e., NNs). The purpose of the attack was to identify relevant perturbations in an adversarial sample to compromise the system, while maintaining the explanation of the sample. The authors presented three cybersecurity-related scenarios where the classifiers were variations of the perceptron. These scenarios targeted a perceptron tasked with detecting malicious PDF documents [45], a perceptron for Android malware classification [12], and an IDS using adversarial AEs [67]. They implemented these attacks using the datasets proposed by the original authors for each scenario, which reinforced the validity and reusability of Kuppa et al.'s [66] results.

Kuppa et al. [65] proposed a general formalization of security problems in XAI. First, they highlighted the motivation of their work by describing five real-life use cases: i) *membership inference attacks* (which involved determining if a sample belonged to the training data), ii) *model extraction attacks* (which aimed at retrieving the actual parameters used in a model), iii) *poisoning attacks* (which targeted the classification performance by corrupting training data), iv) *adversarial examples* (which aimed to evade security classifiers), and v) *counterfactual explanations* (the goal of which was to find the input data points that shared similarities with other entry points but did not yield the same classification results).

Although counterfactual explanations were similar to adversarial examples, they differed in purpose; the latter aimed to evade security classifiers, whereas the former facilitated understanding of a model by providing explanations for it. The authors mathematically defined the attack model for these use cases as well as the explanation methods. They exploited three datasets: leaked passwords from multiple incidents, CIC-IDS2017 [110] for network traffic, and a malware dataset collected from Virusshare [129], combined with benign software collected from various sources. They evaluated their implementation against multiple commercial tools and demonstrated the attacks' ability to evade antivirus solutions.

Cantareira et al. [20] proposed a method for investigating models subjected to adversarial examples. The method used the visual analytic framework to explain adversarial attacks. This method explored layers and weights inside a model to determine which areas were triggered by adversarial examples and allowed to compare data from training data and adversarial examples. The method is as follows. First, it trains two models with different datasets—small CNN for MNIST dataset and MobileNet V2 for ImageNet. Second, it generates adversarial data using projected gradient descent, while selecting random data from the training sample as the background data, to create an adversarial set. Third, it selects an image from the baseline, runs it through the models, and projects its output on a view with the adversarial counterpart. With this, we observed the change in their behavior at each layer of the models, though the effectiveness of the view depends on how representative the background data is.

Zhang et al. [139] argued that DNN interpretations were vulnerable to adversarial attacks. They defined adversarial attacks as Adv^2, which disrupted both the DNN and its interpretation, and succeeded in deliberately designating a prediction and its interpretation. Thus, they could generate adversarial samples, the predictions of which were interpreted as benign samples. Notably, the DNN and its interpretation are nonlinear, which allows an attacker to exploit both. Finally, they explored the possible countermeasures to handle these attacks. They mathematically proved the effectiveness of their proposal, and demonstrated it through experiments performed on the ImageNet [25] and ISIC 2018 challenge dataset [21].

5.3 Privacy

XAI methods can be used to violate the privacy of either the model or data. While several papers have explored the privacy issues of AI algorithms, few have focused on the privacy of XAI methods. Likewise, to protect the privacy of the model users, the explanation model should be unclonable. The literature contains a report on the cloning a model.

Zhao et al. [141] showed that attackers could exploit the flaws of XAI to reconstruct a private image from model explanations. This is called an *image-based model inversion attack*. They also indicated that the image reconstructed based on model prediction alone lacked quality, and that the exploitation of one or more explanations significantly improved the quality. Their experiments also confirmed that XAI methods that provided more explanations resulted in a greater loss of privacy, as they provided richer information that can be exploited by attackers. The authors called for further studies on the tradeoff between explicability and privacy.

Shokri et al. [113] investigated how explainability, in addition to model predictions, contributed to performing membership inference attack, i.e., inferring whether a particular data point was present in the training dataset. They studied several types of explainability models and found that some methods, such as perturbation-based explanations (e.g., *SmoothGrad* [119]), were more robust than backpropagation explanations such as integrated gradient [122], although they produced explanation models with lower quality. They also demonstrated that an attacker could reconstruct most of the dataset from the AI model prediction and XAI results (example-based explanation method). Finally, their experiments showed that i) the correlation between explainability and membership varied with the size of the dataset and that it was easier to execute a membership attack using XAI, on high-dimensional datasets, and ii) that minority data were more likely to be revealed.

Miura et al. [84] proposed a data-free model extraction (DFME) attack called MEGEX. The objective of that study was to clone a model without the initial dataset using both the prediction and explanation of the results. The initial hypothesis for cloning a model is that if the dataset used for training is available, the model can be reproduced by requesting the prediction; however, when the dataset is unknown, methods based on generative models that generate data for requested data to the victim model can be chosen. With MEGEX, the authors used the explanation to train a generative model. They compared their method with an existing method, DFME [127], and evaluated the test-data accuracy of the clone model from the two methods with the CIFAR-10 [64] and SVHN [91] datasets. They achieved a better result with fewer queries to the victim model. The main advantage of this method is its ability to reproduce a model with high accuracy and fewer queries. However, it requires access to the explanation from VanillaGrad.

5.4 Robustness

As we have previously defined the integrity property, where an attacker is attempting to forge explanations, we will discuss here works related to the robustness of explanations. In the following works, robustness is considered variations in explanations induced by variations of the original input.

Kang et al. [56] discussed attacking DNNs and their explanations with adversarial patches. An adversary patch is a correctly localized rectangle that does not hide objects in an image and causes its misclassification. By varying the location and perturbation ratio of the patches, the authors showed that these attacks could i) perturb the result of the DNN, but not those of the XAI model. The model can then determine the patch responsible for the misclassification; ii) perturb the DNN and explainability method, which can no longer detect the patches and designates other regions of the image as responsible for the decision.

The XAI method considered was Grad-CAM, and experiments were performed on ImageNet.

Shi et al. [112] presented a variant of the IFGSM [68] attack, which is in turn based on FGSM [42]. The general idea of the IFGSM is to repeatedly adjust the perturbation direction with a fixed step size. Shi et al. [112] proposed an adaptative FGSM (Ada-FGSM) attack, where the step was dynamically adjusted to improve the performance. They compared their results with the performance of four other models; in terms of success rate and accuracy metrics, Ada-FGSM performed up to 1% better than the second-best model, IFGSM. They also used visualization techniques to follow the gradient evolution of these iterative algorithms and analyzed the results of their experiments. However, their analysis was limited to a few visual examples and was not applied to the entire dataset of their experiment. The generality of the behavior of Ada-FGSM with other data samples is difficult to determine.

Ghorbani et al. [38] indicated that the interpretation of an NN's decision is fragile as a small adversarial perturbation could change it without changing the classification results. Disrupting the interpretation of NNs could harm the trustworthiness of certain applications, e.g., healthcare, where the interpretable decision was relevant as the classification result. They considered two categories of interpretation methods—the first one explained results by identifying the most important features (feature importance methods), and the second one by identifying training samples that contributed the most to the classification result (influence function methods). Adversarial attacks on interpretation methods aim to disrupt the contribution scores assigned to features and training samples by lowering the score of the most important features/training samples. The authors also defined iterative attacks for feature importance methods, and gradient sign-based attacks for influence function methods. To measure the robustness of the different interpretation methods, the authors compared the interpretations (saliency maps) of each method on the original input, and its perturbed version. They used two metrics to measure the similarity between two interpretations—one based on rank correlation, and

the other on the intersection of selected features/training samples identified as important by each interpretation. The authors highlighted that the vulnerability of the interpretation methods stemmed from the high dimensionality of the input instances and highly nonlinear structures of deep networks.

Rieger et al. [99] proposed a defense method against adversarial attacks on explanations. They aggregated multiple explanations methods to reduce the variance of each explanation, thereby enhancing stability against adversarial examples. They explored two scenarios: one where the attacker was unaware of the XAI method used and optimized against the wrong one, and the other where the attacker optimized against the aggregation of XAI methods. In both cases, the attacker had full control over the input and full knowledge (but no control) about the network. In addition, in the second scenario, the attacker knew the parameters and ratios used by the aggregation method. Their results show that, if an attacker optimizes his/her example against one method, the attack does not accurately translate into another method. We assume that an attacker would occasionally optimize against the correct explanation method. In the second scenario, despite the attacker having full knowledge about the target system, the results showed that the proposed aggregation method was resilient against adversarial examples. In the experiments, they used VGG16 [114], a deep CNN and the ImageNet [26] dataset. Although the study explored multiple explanation methods, it omitted the integrated gradient [122] method from the aggregation owing to its computational complexity, thereby indicating a potential scalability issue.

Fenoy et al. [69] proposed a method for evaluating the robustness of XAI algorithms against adversarial noise. They studied the effect of the FGSM adversarial attack on two XAI algorithms: Grad-CAM [108] and SIDU [89]. These algorithms work on images, so the authors proposed to use a portion of 100 natural images from ImageNet. However, to assess the robustness of the algorithms, the ground truth of the explanation is required. Hence, Fenoy et al. [69] collected data from an eye tracker to create a heatmap for each image of the dataset. After collecting all data, they used Grad-CAM and SIDU to obtain a new heatmap, applied an adversarial attack (FGSM) on the data, and measured the Kullback–Leibler divergence between the eye tracker and the XAI algorithms. This method permits assessing and finding highly robust algorithms against adversarial attacks, albeit only on FGSM in Fenoy et al.'s study.

A few papers focused on the mathematical proof of the robustness of XAI methods. For example, Kindermans et al. [59] highlighted the sensitivity of saliency methods (used for XAI) to adversarial attacks. They proposed an axiom called "input variance," which states that these methods must verify to become robust. They mathematically proved the effectiveness of their method and demonstrated it through experiments on the MNIST database.

Malfa et al. [70] developed a robust method to locally explain decisions made by NNs in NLP. They performed mathematical and experimental demonstrations of their proposal on sentiment analysis datasets. They trained fully connected NNs and CNNs for their experiments, and the results showed that the explanations could detect the words in the sentence that influenced the prediction.

Attribution methods assign a score to each feature based on its contribution to the classifier's prediction. Wang et al. [134] showed that attribution methods were vulnerable to adverse perturbations. They modeled the vulnerability of these methods in terms of the *geometry of the targeted model's decision surface* and formalized their robustness as a *local Lipschitz condition on the mapping*. Subsequently, they proposed a *smooth surface regularization* to improve the robustness of all gradient-based attribution methods.

Several detectors of Android malware now exist, but they are generally vulnerable to adversarial examples that evade detection. Melis et al. [83] used gradient-based explanations to evaluate the robustness of a model against such adversarial attacks. They used gradient-based explanations to quantify the *evenness* of feature importance, i.e., how close it is to equal importance for each feature. They also experimentally demonstrated that this uniformity was strongly correlated to the adversarial robustness for the gradient input and integrated gradients methods.

Boopathy et al. [17] showed that attacks that sought classification and explanation evasion would generally disturb the explanation of the original class as well. To detect such attacks, they proposed a metric called l_1 2-class discrepancy, which measured the explanation discrepancy of both the original class (e.g., malicious software) and the target class (e.g., benign software). For nonbinary classification problems, the authors adapted their method by weighting classes according to their importance in prediction. By incorporating this metric into the learning loss, highly robust models could be learned. Their experiments validated the concept and demonstrated the high robustness of the model.

5.5 Evaluation of the explanation

A "good" explanation is arguably difficult to define, as every stakeholder has different needs. In this section, we present studies that provide concrete evaluation of explanations, concerning the properties previously discussed. At the time of writing, the most popular perspective for evaluation was the robustness of the explanation.

Lin et al. [74] highlighted the lack of rigorous and computationally inexpensive evaluation approaches and metrics to quantify the performance of explainability methods. To address this, they evaluated XAI methods based on their

ability to detect a backdoor trigger present in an input and causing its misclassification. A backdoor trigger is a small patch in an image (a yellow square in this paper) that causes the classifier (Trojan model) to misclassify the input. A good explanation should indicate that the reason for such misclassification is precisely the presence of this trigger. The explanation should indicate it as the region that contributed most to this misclassification. To diversify the evaluation measures, the authors considered several variations of the trigger by varying its color, size, and position in an image. They evaluated their proposal by considering several classifiers and different explainability methods on the ImageNet dataset [103].

Hooker et al. [52] proposed a measure of the approximate accuracy of feature importance. They highlighted the challenge of evaluating the reliability of an explanation in the absence of a ground truth. One existing technique is to remove the important features from the input and observe the decrease trend of the classifier accuracy. However, this method has its drawbacks: it does not comply to one of the assumptions in ML that the training and evaluation data come from the same distribution. To address this issue, the authors proposed to evaluate the XAI methods by assessing the decrease trend of the accuracy of a retrained model as the important features are removed. They named this approach "remove and retrain" (ROAR). The authors also performed a large-scale experiment where they choose the model ResNet-50, six estimators of feature importance, and three renowned open-source image datasets (ImageNet [24], Food 101 [18], and Birdsnap [16]). The results reveal that the commonly used base estimators were on par with a random assignment of importance. They concluded that their findings were pertinent to sensitive domains where the accuracy of XAI is crucial.

Adebayo et al. [3] showed that some saliency map (e.g., guided backpropagation and guided Grad-CAM) methods were, in fact, model- and data-independent (such as an edge-detection algorithm). Model independence means that a method is mainly based on data and not on a model; it evolves by comparing with a random model. Data independence entails learning a model on data with completely shuffled labels. Therefore, the explanations created from such saliency maps are irrelevant to understanding the classifier's decision. The results show that visual verification of an explanation alone cannot evaluate an explanation method.

5.6 Discussion

The security of XAI methods has mostly been investigated through the lens of an attacker. We surveyed 27 papers on the security of XAI, but only 6 of them described countermeasures to a security problem. Additionally, most of them addressed the vulnerabilities of DNNs. The crafting of adversarial examples to evade classifiers has proven to be highly accessible in the computer vision field, and to some extent to other fields as well. The topic of fairness will become an important societal issue when automated decisions are widely adopted. Existing works show the fragility of explanations, and attribute it to the high dimensionality and nonlinearities inherent to deep networks.

6 Discussion and future work

In the previous sections, we surveyed scientific publications that either employed explanation methods for cybersecurity use cases or directly investigated the security of explanation methods. Most existing studies target the same specific topics; only a few have broadened their research scope. In this section, we discuss different research avenues that we believe constitute the future direction of secure explanation methods. These topics cover several requirements, such as legal, business efficiency, and performance evaluation.

6.1 From feature space to problem space

A challenge that was not addressed before is the reusability of XAI methods when applied to cybersecurity use cases. The type of cybersecurity data varies across different security applications. Information regarding how security incidents are triggered and attacks are orchestrated may be lost when processing raw security records through ML-based classification models. Pierazzi et al. [96] provided a formal definition of the transformation mapping the problem space to the feature space from the perspective of adversarial attacks against malware detection algorithms. Manipulations over the feature space that successfully flip a classifier's decision output may violate the feasibility constraint posed by the problem space. For example, malware detection generally employs the term frequency-inverse document frequency (TF-IDF) vectors of n-grams of opcodes and/or dynamic analysis traces as feature representation. Modifying the TF-IDF feature vector may easily change the classifier's output. However, determining the changes in the raw opcodes/system calls that can cause variations in the TF-IDF feature representation remains a challenging task. This is intrinsically an ill-posed reverse problem. Further, the low-level codes/system calls of a malware sample are organized to deliver malicious functions. Blind modification of any code segment might render the malware sample inexecutable. The feasibility constraints, i.e., coding syntax and code design of malware samples, implicitly limit the possible modifications in the problem and feature spaces. However, these constraints are not readily available and difficult to encode in a computable manner. In modern ML models, such as DNNs, mapping from the problem and

feature spaces can be complex. Owing to the multilayered and highly nonlinear transformation embedded in a DL architecture, the association between the raw attributes given in the input and the embedding vectors derived at the deep layers cannot always be presented analytically. Therefore, we observed a gap between the interpretation given in the problem space and that provided in the feature space in most ML methods, especially in DL models. Although SHAP, LIME, and layer relevance propagation-based XAI methods can be used to unveil important features encoded by a DL model in classification, the link between the importance of the encoded embedding features with the contribution of the raw attributes given in the problem space is difficult to be reversibly recovered.

6.2 Privacy vs. explainability tradeoff

As discussed in Section 5.3, few papers have addressed the privacy challenges for XAI methods. Existing works have demonstrated that privacy and the design of an effective XAI method can be contradictory, and that several XAI methods can be useful but not secure. The two cases that have been explored are the *"model inversion attack"* and the *"membership inference attack"*. However, Liu et al. [75] covered several other types of privacy attacks orchestrated on AI models, such as *"feature estimation attack"* (learning some features from the training dataset), *"model extraction attack"* (learning an approximation of the model), and *"model memorization attack"* (retrieving the exact values of features). These attacks can also be extended to explainability methods used to interpret AI models, as indicated in [113, 141].

Most studies on privacy in AI focused on highlighting the possible attacks rather than proposing solutions to them. Nevertheless, a few solutions indeed have been proposed, as indicated in [75], e.g., i) encryption training data and ML model, ii) reduction of the accuracy of data and model available to an attacker using obfuscation mechanisms, such as noise addition to the output classifier and iii) use of aggregation techniques in the case of collaborative learning to keep collaborating models and datasets private.

However, these methods must be adapted to the XAI context and tested to measure their effectiveness in preserving the privacy of models and data.

In conclusion, XAI adds an additional layer to the attack surface of the AI pipeline; therefore, its use must be carefully considered—is it wise to use an XAI method to understand an AI model, at the risk of making the system vulnerable, or should it be dispensed with to reduce the attack surface? Thus, the choice of an XAI method offering the best compromise between explanatory quality and confidentiality is crucial.

6.3 Reinforcing explainability with external data

Various methods using XAI have been reported. XAI offers explanations on the decisions made by AI models. Methods thus far have focused on generating explanations solely based on data and algorithms used by AI models. However, external data that have not been used by security models can be used to reinforce the explainability of alerts. When security operators recognize a malware activity on the Internet, they frequently search in external data, such as honeypots analysis results and/or threat intelligence reports, to confirm their discovery.

Takahashi et al. [125] employed this approach to reinforce the explainability of AI models. They established several security-analysis models, including malware traffic detection, sandbox analysis, code analysis, threat intelligence search, and Darknet traffic analysis models. Various security operations are automated with comprehensive data. For example, when malware traffic is detected by a traffic analysis model (AI-based), the system runs other models and collects comprehensive information, including first name, family name, behavior, exploited vulnerabilities, and target devices. By reviewing this information, users can confirm that the malware traffic analysis model has detected a real threat.

Unlike other XAI studies in the AI domain, XAI for cybersecurity research can utilize assorted security-related data. Therefore, these data could be used to explain the decisions made by security AI models and could reinforce the explainability of the AI models. This type of study should be furthered.

7 Conclusion

As with any other computer science field, Cybersecurity has been widely studied under the scope of AI. XAI is gaining momentum and might become a legal requirement for various service providers. We surveyed the existing XAI literature related to cybersecurity from two perspectives—XAI for cybersecurity tasks and security of XAI methods. From the first perspective, although the existing literature already has a considerable number of scientific papers, we found a lack of consideration for real scenarios in their approaches. We believe that meaningful scientific contributions should go beyond the application of an XAI method to a cybersecurity dataset, and attempts must be made to bridge the gap between the fields of AI and cybersecurity. From the second perspective, we divided the security issues into multiple classes and addressed the existing literature regarding the attack surface, current attack vectors, and potential countermeasures. The state of the art reveals a lack of countermeasures for the defense of XAI methods, which is also

reflected in CV, where heatmaps and saliency maps are easily compromised. We highlighted several research avenues to improve the security of explainable methods, covering both practical aspects such as privacy concerns and ethical aspects, including fairness and fairwashing. We conclude this survey by reaffirming that AI will be a major actor in enforcing business policies and assisting with important decision-making matters. As such, XAI should guarantee fair, clear, and unbiased treatment.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12243-022-00926-7.

Declarations

Conflict of interest Not applicable.

References

1. 2018 reform of EU data protection rules. European Commission. May 25, 2018 (visited on 07/25/2022). https://ec.europa.eu/info/sites/default/files/data-protection-factsheet-changes_en.pdf
2. Adadi A, Berrada M (2018) Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). In: IEEE Access, vol 6, pp 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052
3. Adebayo J et al (2018) Sanity checks for saliency maps. In: Advances in neural information processing systems, p 31
4. Aguilar DL et al (2022) Towards an interpretable autoencoder: A decision tree-based autoencoder and its application in anomaly detection. In: IEEE transactions on dependable and secure computing
5. Ahmad MW, Reynolds J, Rezgui Y (2018) Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. In: Journal of cleaner production, vol 203, pp 810–821
6. Ahmed M et al (eds) (2022) Explainable artificial intelligence for Cyber security. Springer International Publishing, Berlin. https://doi.org/10.1007/978-3-030-96630-0
7. Aïvodji U et al (2019) Fairwashing: the risk of rationalization. In: arXiv:1901.09749
8. Alenezi R, Ludwig SA (2021) Explainability of cybersecurity threats data using SHAP. In: 2021 IEEE symposium series on computational intelligence (SSCI). IEEE, pp 01–10
9. Alqaraawi A et al (2020) Evaluating saliency map explanations for convolutional neural networks: a user study. In: Proceedings of the 25th international conference on intelligent user interfaces, pp 275–285
10. Anders C et al (2020) Fairwashing explanations with off-manifold detergent. In: International conference on machine learning. PMLR, pp 314–323
11. Antwarg L et al (2021) Explaining anomalies detected by autoencoders using Shapley additive explanations. In: Expert systems with applications, vol 186, p 115736
12. Arp D et al (2014) Drebin: Effective and explainable detection of android malware in your pocket. In: Ndss, vol 14, pp 23–26
13. Arrieta AB et al (2020) Explainable artificial intelligence (XAI): Concepts, Taxonomies, Opportunities and challenges toward responsible AI. In: Inf Fusion, vol 58, pp 82–115
14. Asuncion A, Newman D (2007) UCI machine learning repository. Accessed: 2022-03-25. http://archive.ics.uci.edu/ml
15. Bach S et al (2015) On pixel-wise explanations for non-linear classifer decisions by layerwise relevance propagation. In: PloS One, vol 10.7, p e0130140
16. Berg T et al (2014) Birdsnap: Large-scale finegrained visual categorization of birds. In: 2014 IEEE conference on computer vision and pattern recognition, pp 2019–2026. https://doi.org/10.1109/CVPR.2014.259
17. Boopathy A et al (2020) Proper network interpretability helps adversarial robustness in classification. In: International conference on machine learning. PMLR, pp 1014–102
18. Bossard L, Guillaumin M, Gool LV (2014) Food-101 - mining discriminative components with random forests. In: ECCV
19. Breiman L (2001) Random forests. In: Machine learning, vol 45.1, pp 5–32
20. Cantareira GD, Mello RF, Paulovich FV (2021) Explainable adversarial attacks in deep neural networks using activation profiles. In: arXiv:2103.10229
21. Codella N et al (2019) Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). In: arXiv:1902.03368
22. CSE-CIC-IDS2018 on AWS. Accessed: 2022-03-25 (2018) https://www.unb.ca/cic/datasets/ids-2018.html
23. Dellermann D et al (2019) Hybrid intelligence. In: Business & information systems engineering, vol 61.5, pp 637–643
24. Deng J et al (2009) ImageNet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
25. Deng J et al (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee, pp 248–255
26. Deng J et al (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee, pp 248–255
27. Dimanov B et al (2020) You Shouldn't Trust Me: Learning models which conceal unfairness from multiple explanation methods. In: SafeAI@AAAI
28. Dombrowski A-K et al (2019) Explanations can be manipulated and geometry is to blame. In: Advances in neural information processing systems, p 32
29. Došilović FK, Brčcić M, Hlupić N (2018) Explainable artificial intelligence: A survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), pp 0210–0215. https://doi.org/10.23919/MIPRO.2018.8400040
30. Van Eck D, McAdams DA, Vermaas PE (2007) Functional decomposition in engineering: a survey. In: International design engineering technical conferences and computers and information in engineering conference, vol 48043, pp 227–236

31. Elson J et al (2007) Asirra: A CAPTCHA that exploits interest-aligned manual image categorization. In: CCS, vol 7, pp 366–374
32. Fan Y et al (2021) Understanding update of machine-learning-based malware detection by clustering changes in feature attributions. In: International workshop on security. Springer, pp 99–118
33. Faraway JJ (2016) Extending the linear model with R. Chapman and Hall/CRC. https://doi.org/10.1201/9781315382722
34. Farrugia D et al (2021) A real-time prescriptive solution for explainable Cyber-Fraud detection within the igaming industry. In: Sn computer science, p 2
35. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. In: Annals of statistics, pp 1189–1232
36. Galkin F et al (2018) Human microbiome aging clocks based on deep learning and tandem of permutation feature importance and accumulated local effects. In: BioRxiv, p 507780
37. Galli A et al (2021) Reliability of explainable artificial intelligence in adversarial perturbation scenarios. In: International conference on pattern recognition. Springer, pp 243–256
38. Ghorbani A, Abid A, Zou J (2019) Interpretation of neural networks is fragile. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 3681–3688
39. Giudici P, Raffinetti E (2022) Explainable AI methods in cyber risk management. In: Quality and reliability engineering international, vol 38.3, pp 1318–1326
40. Giudici P, Raffinetti E (2021) Shapley-Lorenz eXplainable artificial intelligence. In: Expert systems with applications, vol 167, p 114104
41. Goldstein A et al (2015) Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. In: Journal of computational and graphical statistics, vol 24.1, pp 44–65
42. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. In: arXiv:1412.6572
43. Gulmezoglu B (2021) XAI-based microarchitectural side-channel analysis for website fingerprinting attacks and defenses. In: IEEE transactions on dependable and secure computing
44. Gunning D, Aha D (2019) DARPA's explainable artificial intelligence (XAI) program. In: AI magazine, vol 40.2, pp 44–58
45. Guo W et al (2018) Lemna: Explaining deep learning based security applications. In: proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 364–379
46. Hagras H (2018) Toward Human-Understandable Explainable AI. In: Computer, vol 51.9, pp 28–36. https://doi.org/10.1109/MC.2018.3620965
47. Hanif A, Zhang X, Wood S (2021) A survey on explainable artificial intelligence techniques and challenges. In: 2021 IEEE 25th international enterprise distributed object computing workshop (EDOCW), pp 81–89. https://doi.org/10.1109/EDOCW52865.2021.00036
48. Hastie T et al (2009) The elements of statistical learning: data mining, inference, and prediction, vol 2. Springer, Berlin
49. Heo J, Joo S, Moon T (2019) Fooling neural network interpretations via adversarial model manipulation. In: Advances in neural information processing systems, p 32
50. Holzinger A et al (2018) Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable AI. In: International cross-domain conference for machine learning and knowledge extraction. Springer, pp 1–8
51. Hong S et al (2015) Online tracking by learning discriminative saliency map with convolutional neural network. In: International conference on machine learning. PMLR, pp 597–606
52. Hooker S et al (2019) A benchmark for interpretability methods in deep neural networks. In: Advances in neural information processing systems, p 32
53. Iadarola G et al (2021) Towards an interpretable deep learning model for mobile malware detection and family identification. In: Computers & Security, vol 105, p 102198
54. Islam SR et al (2019) Domain knowledge aided explainable artificial intelligence for intrusion detection and response. In: arXiv:1911.09853
55. Kirchner L, Larson J, Mattu S, Angwin J (2020) Propublica Recidivism Dataset. https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis. Accessed 01 Aug 2022
56. Kang H, Kim H et al (2021) Robust adversarial attack against explainable deep classification models based on adversarial images with different patch sizes and perturbation ratios. In: IEEE Access, vol 9, pp 133049–133061
57. Karn RR et al (2020) Cryptomining detection in container clouds using system calls and explainable machine learning. In: IEEE transactions on parallel and distributed systems, vol 32.3, pp 674–691
58. Khan IA et al (2022) XSRU-IoMT: Explainable simple recurrent units for threat detection in Internet of Medical Things networks. In: Future generation computer systems, vol 127, pp 181–193
59. Kindermans P-J et al (2019) The (un) reliability of saliency methods. In: Explainable AI: Interpreting, explaining and visualizing deep learning. Springer, pp 267–280
60. Kingsford C, Salzberg SL (2008) What are decision trees?. In: Nature Biotechnology, vol 26.9, pp 1011–1013
61. Kleinbaum DG et al (2002) Logistic regression. Springer, New York
62. Kohavi R, Becker B (2020) UCI - Adult Dataset. https://archive.ics.uci.edu/ml/datasets/adult
63. Koroniotis N et al (2019) Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. In: Future generation computer systems. issn: 0167-739X, vol 100, pp 779–796. https://doi.org/10.1016/j.future.2019.05.041
64. Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images
65. Kuppa A, Le-Khac N-A (2021) Adversarial xai methods in cybersecurity. In: IEEE transactions on information forensics and security, vol 16, pp 4924–4938
66. Kuppa A, Le-Khac N-A (2020) Black box attacks on explainable artificial intelligence (XAI) methods in cyber security. In: 2020 international joint conference on neural networks (IJCNN). IEEE, pp 1–8
67. Kuppa A et al (2019) Finding rats in cats: Detecting stealthy attacks using group anomaly detection. In: 2019 18th IEEE international conference on trust, security and privacy in computing and communications/13th IEEE international conference on big data science and engineering (TrustCom/BigDataSE). IEEE, pp 442–449
68. Kurakin A, Goodfellow IJ, Bengio S (2018) Adversarial examples in the physical world. In: Artificial intelligence safety and security. Chapman and Hall/CRC, pp 99–112
69. Ciontos A, Fenoy LM (2020) Performance evaluation of explainable ai methods against adversarial noise
70. La Malfa E et al (2021) On guaranteed optimal robust explanations for NLP models. In: arXiv:2105.03640
71. Lashkari AH et al (2017) Characterization of tor traffic using time based features. In: ICISSp, pp 253–262
72. Le Merrer E, Trédan G (2020) Remote explainability faces the bouncer problem. In: Nature machine intelligence, vol 2.9, pp 529–539

73. Li L-J, Fei-Fei L (2007) What, where and who? classifying events by scene and object recognition. In: 2007 IEEE 11th international conference on computer vision. IEEE, pp 1–8
74. Lin Y-S, Lee W-C, Celik ZB (2020) What do you see? Evaluation of explainable artificial intelligence (XAI) interpretability through neural backdoors. In: arXiv:2009.10639
75. Liu B et al (2021) When machine learning meets privacy: A survey and outlook. In: ACM Computing Surveys (CSUR), vol 54.2, pp 1–36
76. Liu H et al (2021) FAIXID: a framework for enhancing ai explainability of intrusion detection results using data cleaning techniques. In: Journal of network and systems management, vol 29.4, pp 1–30
77. Longo L et al (2020) Explainable artificial intelligence: Concepts, applications, research challenges and visions. In: International cross-domain conference for machine learning and knowledge extraction. Springer, pp 1–16
78. Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Advances in neural information processing systems, p 30
79. Mahbooba B et al (2021) Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model. In: Complexity, 2021
80. Mahdavifar S, Alhadidi D, Ghorbani A et al (2022) Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder. In: Journal of network and systems management, vol 30.1, pp 1–34
81. Mamun MSI et al (2016) Detecting malicious urls using lexical analysis. In: International conference on network and system security. Springer, pp 467–482
82. Marino DL, Wickramasinghe CS, Manic M (2018) An adversarial approach for explainable ai in intrusion detection systems. In: IECON 2018-44th annual conference of the IEEE industrial electronics society. IEEE, pp 3237–3243
83. Melis M et al (2022) Do gradient-based explanations tell anything about adversarial robustness to android malware?. In: International journal of machine learning and cybernetics, vol 13.1, pp 217–232
84. Miura T, Hasegawa S, Shibahara T (2021) MEGEX: Data-free model extraction attack against gradient-based explainable AI. In: arXiv:2107.08909
85. Molnar C (2018) A guide for making black box models explainable. In: https://christophm.github.io/interpretable-ml-book. Accessed 01 Aug 2022
86. MontazeriShatoori M et al (2020) Detection of doh tunnels using time-series classification of encrypted traffic. In: 2020 IEEE intl conf on dependable, autonomic and secure computing, intl conf on pervasive intelligence and computing, intl conf on cloud and big data computing, intl conf on cyber science and technology congress (DASC/PiCom/CBDCom/CyberSciTech). IEEE, pp 63–70
87. Moustafa N (2019) New generations of internet of things datasets for cybersecurity applications based machine learning: TON IoT datasets. In: Proceedings of the eResearch Australasia Conference. Brisbane, Australia, pp 21–25
88. Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 military communications and information systems conference (MilCIS). IEEE, pp 1–6
89. Muddamsetty SM et al (2022) Visual explanation of black-box model: Similarity difference and uniqueness (SIDU) method. In: Pattern recognition, vol 127, p 108604
90. Müller J, Shoemaker CA, Piché R (2013) SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. In: Computers & operations research, vol 40.5, pp 1383–1400
91. Netzer Y et al (2011) Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning, p 2011
92. Nguyen A, Yosinski J, Clune J (2019) Understanding neural networks via feature visualization: A survey. In: Explainable AI: interpreting, explaining and visualizing deep learning. Springer, pp 55–76
93. Pagès J (2014) Multiple factor analysis by example using R. CRC Press
94. Paredes J et al (2021) On the importance of domainspecific explanations in AI-based cybersecurity systems (Technical Report). In: arXiv:2108.02006
95. Pedreshi D, Ruggieri S, Turini F (2008) Discrimination-aware data mining. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, pp 560–568
96. Pierazzi F et al (2020) Intriguing properties of adversarial ML Attacks in the problem space. English. In: 2020 IEEE symposium on security and privacy. issn: 2375–1207, pp 1332–1349. https://doi.org/10.1109/SP40000.2020.00073
97. Ribeiro MT, Singh S, Guestrin C (2016) Why should i trust you? Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144
98. Ribeiro MT, Singh S, Guestrin C (2018) Anchors: High-precision modelagnostic explanations. In: Proceedings of the AAAI conference on artificial intelligence, vol 32, p 1
99. Rieger L, Hansen LK (2020) A simple defense against adversarial attacks on heatmap explanations. In: arXiv:2007.06381
100. Rish I et al (2001) An empirical study of the naive Bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence, vol 3, pp 41–46
101. Rosenblatt F (1958) The Perceptron: A probabilistic model for information storage and organization in the brain. In: Psychological review, pp 65–386
102. Roshan K, Zafar A (2021) Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation(SHAP). In: arXiv:2112.08442
103. Russakovsky O et al (2015) Imagenet large scale visual recognition challenge. In: International journal of computer vision, vol 115.3, pp 211–252
104. Samek W, Wiegand T, Müller K-R (2017) Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. In: arXiv:1708.08296
105. Samek W et al (2021) Explaining deep neural networks and beyond: A review of methods and applications. In: Proceedings of the IEEE, vol 109.3, pp 247–278
106. Sarhan M, Layeghy S, Portmann M (2021) Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection
107. Sarhan M, Layeghy S, Portmann M (2021) Towards a standard feature set for network intrusion detection system datasets. In: Mobile networks and applications, pp 1–14
108. Selvaraju RR et al (2017) Grad-cam: Visual explanations from deep networks via gradientbased localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
109. Shahid MR, Debar H (2021) CVSSBERT: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In: 2021 20th IEEE international conference on machine learning and applications (ICMLA). IEEE, pp 1600–1607
110. Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSp, vol 1, pp 108–116

111. Shaukat K et al (2020) A survey on machine learning techniques for cyber security in the last decade. In: IEEE Access, vol 8, pp 222310–222354
112. Shi Y et al (2020) Adaptive iterative attack towards explainable adversarial robustness. In: Pattern recognition, vol 105, p 107309
113. Shokri R, Strobel M, Zick Y (2021) On the privacy risks of model explanations. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pp 231–241
114. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. In: arXiv:1409.1556
115. Sinha S et al (2021) Perturbing inputs for fragile interpretations in deep natural language processing. In: arXiv:2108.04990
116. Slack DZ et al (2021) Reliable Post hoc Explanations: Modeling Uncertainty in Explainability. In: Beygelzimer A et al (eds) Advances in neural information processing systems. https://openreview.net/forum?id=rqfq0CYIekd. Accessed 01 Aug 2022
117. Slack D et al (2021) Counterfactual explanations can be manipulated. In: Advances in neural information processing systems, vol 34, pp 62–75
118. Slack D et al (2020) Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM conference on AI, Ethics, and Society, pp 180–186
119. Smilkov D et al (2017) SmoothGrad: removing noise by adding noise. In: arXiv:1706.03825
120. Smutz C, Stavrou A (2012) Malicious PDF detection using metadata and structural features. In: Proceedings of the 28th annual computer security applications conference, pp 239–248
121. Srivastava G et al (2022) XAI for Cybersecurity: State of the Art, Challenges, Open Issues and Future Directions. In: arXiv:2206.03585
122. Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: International conference on machine learning. PMLR, pp 3319–3328
123. Szczepański M et al (2020) Achieving explainability of intrusion detection system by hybrid oracle-explainer approach. In: 2020 international joint conference on neural networks (IJCNN). IEEE, pp 1–8
124. Tahmina Z, Rezvy S, Yuan L et al (2022) An explainable ai-based intrusion detection system for DNS over HTTPS (DoH) Attacks. In: Techrxiv
125. Takahashi T et al (2021) Designing comprehensive cyber threat analysis platform: Can we orchestrate analysis engines?. In: 2021 IEEE international conference on pervasive computing and communications workshops and other affiliated events (PerCom Workshops). IEEE, pp 376–379
126. Tavallaee M et al (2009) A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. Ieee, pp 1–6
127. Truong J-B et al (2021) Data-free model extraction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4771–4780
128. Vigano L, Magazzeni D (2020) Explainable security. In: 2020 IEEE European symposium on security and privacy workshops (EuroS&PW). IEEE, pp 293–300
129. Virus Share: Virus Report Sharing. Accessed: 2022-03-22. https://virusshare.com
130. Wachter S, Mittelstadt B, Russell C (2017) Counterfactual explanations without opening the black box: Automated decisions and the GDPR. In: Harv. JL & Tech, vol 31, p 841
131. Wali S, Khan I (2021) Explainable AI and random forest based reliable intrusion detection system. In: https://doi.org/10.36227/techrxiv.17169080.v1
132. Wang M et al (2020) An explainable machine learning framework for intrusion detection systems. In: IEEE Access, vol 8, pp 73127–73141
133. Wang S et al (2016) Trafficav: An effective and explainable detection of mobile malware behavior using network traffic. In: 2016 IEEE/ACM 24th international symposium on quality of service (IWQoS). IEEE, pp 1–6
134. Wang Z et al (2020) Smoothed geometry for robust attribution. In: Advances in neural information processing systems, vol 33, pp 13623–13634
135. Xu F et al (2019) Explainable AI: A brief survey on history, research areas, approaches and challenges. In: CCF international conference on natural language processing and Chinese computing. Springer, pp. 563–574
136. Zeng X, Martinez T (2001) Distribution-balanced stratified cross-validation for accuracy estimation. In: Journal of experimental & theoretical artificial intelligence vol 12. https://doi.org/10.1080/095281300146272
137. Zeng Z et al (2015) A novel feature selection method considering feature interaction. In: Pattern recognition, vol 48.8, pp 2656–2666
138. Zhang Q et al (2018) Interpreting cnn knowledge via an explanatory graph. In: Proceedings of the AAAI conference on artificial intelligence, vol 32, p 1
139. Zhang X et al (2020) Interpretable deep learning under fire. In: 29th fUSENIXg Security Symposium (fUSENIXg Security 20)
140. Zhao X et al (2021) BayLIME: Bayesian local interpretable model-agnostic explanations. In: UAI
141. Zhao X et al (2021) Exploiting explanations for model inversion attacks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 682–692
142. Zolanvari M et al (2019) Machine learning-based network vulnerability analysis of industrial Internet of Things. In: IEEE internet of things journal, vol 6.4, pp 6822–6834
143. Zolanvari M et al (2021) TRUST XAI: Model-agnostic explanations for AI With a Case Study on IIoT Security. In: IEEE internet of things journal