electronics

MDPI

Article

On the Need for Collaborative Intelligence in Cybersecurity

Trevor Martin

Machine Intelligence Unit, Engineering Maths Department, University of Bristol, Bristol BS8 1UB, UK; trevor.martin@bristol.ac.uk

Abstract: The success of artificial intelligence (and particularly data-driven machine learning) in classifying and making predictions from large bodies of data has led to an expectation that autonomous AI systems can be deployed in cybersecurity applications. In this position paper we outline some of the problems facing machine learning in cybersecurity and argue for a collaborative approach where humans contribute insight and understanding, whilst machines are used to gather, filter and process data into a convenient and understandable form. In turn this requires a convenient representation for exchanging information between machine and human, and we argue that graded concepts are suitable, allowing summarisation at multiple levels of discernibility (granularity). We conclude with some suggestions for developing a hierarchical and graded representation.

Keywords: AI in cybersecurity; collaborative intelligence; graded knowledge representation; formal concept analysis; ontology; knowledge graph

1. Introduction

Recent developments have pushed artificial intelligence to the forefront of many applications in areas as diverse as product and service recommendation, autonomous and partially autonomous vehicles, healthcare, finance, voice-driven interfaces/interactive systems, telecom routing, and fraud detection, amongst many others.

This development has been made possible by vast increases in our ability to store, transmit and process data, underpinned by a similarly large leap in the number and capabilities of devices able to gather data, and the general population's acceptance (or ignorance) of the data gathering processes. It is worth pointing out that popular use of the term "artificial intelligence" nowadays frequently refers to the sub-area of statistical or data-driven machine learning that is able to capture patterns in the data and use them to make decisions (often predictions or classifications) in new cases. We use the term in the broader sense, to include knowledge-based as well as data-driven software systems.

AI has been identified as having the potential to change both sides of the cyber-security landscape, both attack and defence. The rapid increase in data-driven AI classifiers and decision-makers is a double-edged sword—it can create more powerful and more adaptive defences, but can also increase the sophistication of an attack.

However, it is important to note that AI (and in particular, data-driven machine learning) is not a "silver bullet" solution to the problems of detecting anomalous activity in cybersecurity. The strength of machine-learning tools is finding activity that is similar to something previously seen, without the need to precisely describe that activity; cybersecurity problems generally involve an adversary who is trying to subvert a system in a novel way, using methods that are new or not previously detected. Deep learning systems are vulnerable to so-called adversarial attacks, where a carefully crafted input can lead to completely incorrect classifications. Great care is needed in ensuring that training data is clean and represents the whole space.

It is difficult to completely define an anomaly in the context of cyber-security—an anomaly is an event (or sequence of events) that is not necessarily rare, but that does not conform to expected behaviour and possibly constitutes a threat. Of course, this begs the

check for updates

Citation: Martin, T. On the Need for Collaborative Intelligence in Cybersecurity. *Electronics* **2022**, *11*, 2067. https://doi.org/10.3390/electronics11132067

Academic Editor: Aryya Gangopadhyay

Received: 26 May 2022
Accepted: 29 June 2022
Published: 30 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

question of what constitutes expected behaviour—frequently, a baseline of data is taken as normal behaviour, and subsequent data is examined for anomalies. If the initial dataset can be guaranteed clean (and representative), this approach is valid, but if it is possible the initial data could contain "anomalous" instances then it risks classifying malicious events as normal. The boundaries of normal behaviour are rarely clear-cut, and may change with time—particularly in cyber-security applications when adversaries often aim to avoid raising suspicion, and can adapt to updated defences. The key to a good cyberattack is undetectability—this distinguishes anomaly detection in cybersecurity data from most applications of anomaly detection, since the presence of an adversary may mean that baseline data can rarely be guaranteed clean.

We therefore consider that fully autonomous cyberdefence is neither likely nor desirable. Whilst many routine tasks can be automated, the overall security of a network or system is an evolving problem that is not amenable to standard machine learning tasks. Indeed, the paucity of available cyberattack data actively works against data-driven machine learning. The possibility of inadvertently releasing personal details (even in anonymised records) is a strong motivation to withhold data. In addition, there is an imperative to stop a cyberattack when it is detected, rather than allowing it to continue so that representative data can be gathered. There are datasets showing simulated attacks but these are somewhat artificial and not necessarily representative of real attacks.

Our view is that collaborative intelligence—a symbiotic combination of humans and machines—is a better approach than autonomous AI to monitor and proactively control the behaviour of a complex system such as a computer network. Even when autonomous decisions are necessary to cope with the pace of events, retrospective analysis demands human involvement. The idea of collaborative intelligence can be traced back to multi-agent systems where computational tasks were distributed amongst processors—either in a heterogeneous manner where each component had a specific "expertise" or in a homogeneous fashion, where a problem was divided into similar, but smaller, sub-problems, processed separately. By allowing "agents" to be human or machine, we arrive at a collaborative intelligence approach. This relies on simple, effective and efficient communication of information between the "processing components", i.e., computers and human analysts. A key feature in enhancing human understanding of large scale data is the notion of summarisation. By combining many values (or objects) into a few entities, concise summaries enable analysts to gain insight into bulk patterns and focus on the mechanisms underlying relations in the data, leading to greater understanding of current data and prediction of future data.

In the remainder of this position paper, we outline some of the key background material and suggest that use of graded concepts is a promising approach to information sharing between humans and machines.

2. The Use of AI Techniques for Anomaly Detection

Anomaly detection predates AI, and is often studied as a sub-field of statistics. Frequently, anomalies are equated with outliers and a variety of statistical techniques exist to detect outliers (and/or noise) in data, both static and dynamic (time-series).

It is common to label as outliers any data points that deviate significantly from common statistical properties such as mean, median, etc., or points that fall into particular quantiles. Two commonly quoted definitions of an outlier are:

> *" . . . an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism"* [1]

and

> *" . . . an observation or subset of observations that appears to be inconsistent with the rest of the set of data".* [2]

The difficulty of detecting anomalies/outliers is often increased in streaming applications [3], where, typically, data can only be viewed once, and data volume/velocity may

mean that detailed analysis is not possible. Additional problems arise from so-called "drift and shift" where properties such as mean (within a specified window) can vary systematically over time, e.g., daily temperatures might exhibit significant seasonal variation, complicating the detection of anomalous values.

Anomalies can be sub-divided into

- Single anomalies (one data point stands out from the rest);
- Contextual anomalies (conditional) where a data point is not unusual in itself, but is not expected in the surrounding set of data—particularly in a time series;
- Collective anomalies where a set of data points appear to be unusual.

The use of anomaly detection for cyber-security applications is obvious, and goes back to at least the 1980s [4], where statistical profiles were proposed to identify anomalous behaviour, and rule-based actions were taken on detection of anomalies.

2.1. What Is an Anomaly

It is worth pointing out the (possibly obvious) fact that an anomaly is not just an example that differs from others. In a sense, all distinct examples are unique, and as more features of the data are considered, there are more ways to distinguish any given instance. To take a trivial example, consider the list "*hen, sheep, cow, dog*" and identify the anomaly (odd one out). There are plausible reasons for identifying any member of the list as anomalous—*hen* is the only bird, *dog* is the only carnivore, *cow* is the only word with its letters in alphabetic order, *sheep* is the only 5 letter word, etc. This may seem a contrived example, but there is an important point here—given a set of examples in a sufficiently rich feature space, it is normally possible to make a case for any example being out of the ordinary and hence anomalous.

There are several good surveys of the anomaly detection problem, both from the general perspective [5,6] and from the more specialised viewpoint of cyber security [7]. For the statistician, an anomaly is an outlier but not all outliers are anomalies. A common statistical starting point (e.g., [8]) is to assume that data is produced by an underlying random process and to distinguish outliers (data produced by the random process, with low probability) from anomalies (data produced by another process). Of course, this is not applicable if the data cannot be modelled as a random process, or if the process cannot be known with sufficient accuracy to distinguish between data arising from the model with low probability and data that is not from the model.

2.2. Artificial Intelligence and Anomaly Detection

Anomaly detection is a natural application for data-driven machine learning. Two popular approaches are

- one-class statistical learning—if we view anomalies as rare events, it makes sense to attempt to learn the patterns of normal behaviour and treat anything outside this class as an anomaly—see [9] for an example. We would argue that the one-class approach is not always applicable for cyber-security applications—it essentially splits data into normal and abnormal, but attempts to learn the classifier only from examples of normal data. Caution is required, since a typical machine learning program needs a large number of representatives of all classes that are to be detected, and when there is only one "normal" class, the "abnormal" class may consist of several disjoint subclasses (including some with no available examples).
- techniques based on deep-learning, particularly use of reconstruction error to indicate anomalies. Essentially this relies on building an auto-encoder using a dataset the defines normal behaviour, and then running the auto-encoder with new data. For data close to the original set, the auto-encoder will produce very similar data as a good approximation to the input; if the predicted input was a long way from the actual input then actual input is likely to be anomalous. A broader overview of "deep anomaly detection" is presented in [10] which includes a section focused on anomalies

in intrusion detection systems and a list of datasets. Problems related to computational complexity and noisy/unbalanced training data are acknowledged, as is the difficulty of retraining to cope with changes in the data stream. Explanation is not considered in the survey. See also [11] for a view of explainable deep learning in process monitoring

Evangelou and Adams [12] present a framework for cybersecurity anomaly detection based on NetFlow records, where an anomaly is defined as an increased number of connection to other hosts, relative to a baseline of "normal" behaviour. Scalability of the approach is not clear—the study is based on data from the Imperial College network which is quoted as having in the region of 40,000 devices connected to it each day; the study rests on analysing 55 randomly selected devices.

Duan et al. [13] apply reinforcement learning to detect unexpected events in log sequences from distributed database and distributed processing applications. These have some similarity to cyber-security logs, although the approach relies on the notion of an event sequence being defined and detectable. This is generally the case in database transactions but may be less well-defined in cyber-security logs.

Provision of representative labelled data is a major issue for many approaches in cyber-security—as mentioned above, a competent adversary may be able to pollute normal data, and once an anomaly (intrusion, etc.) is detected, good practice aims to block the vulnerabilities identified as soon as possible, so that there is limited opportunity to gather examples of the anomalous incident [14]. The lack of data is not just a problem for machine learning applications that rely on adequate sources of data for training; it also causes problems for validation and comparison of different approaches to anomaly detection. As discussed in [15], confidentiality is the main reason—real-world data can reveal highly sensitive information, even when it is seemingly sanitised.

Sommer and Paxson [15] also identify problems in using machine learning for anomaly detection (particularly network intrusions):

> *"despite extensive academic research one finds a striking gap in terms of actual deployments of such systems: compared with other intrusion detection approaches, machine learning is rarely employed in operational "real world" settings".*

Amongst the difficulties identified by these authors are the high cost of errors, lack of adequate training and testing data, and a semantic gap between classification results and operational interpretation (that is to say, once an event is identified as anomalous, there is no indication whether it is an isolated incident arising from user error, etc., or is part of an attack and requires action). The authors also highlight the possibility that in some circumstances it might be quicker and easier to look for a simple, non-machine learning approach.

2.3. Explainability in Anomaly Detection

The ubiquity of data-driven intelligent systems has led to a demand for so-called "explainability" [16]. The vast majority of automated decisions are made on the basis of large datasets and complex internal processing that can be incomprehensible even to domain experts. In many cases we need to understand why a particular output is given, rather than just focus on the accuracy of the output. The DARPA initiative [17] is widely credited with sparking much of the research in the area of statistical machine learning, although the broader area of explainability in AI has a longer history, going back at least to the second AI boom of the 1980's.

There are a number of good surveys and overviews in explainable AI, including [18–20]).

Explainability is a key feature of anomaly detection—whether AI based or not—due to the need to understand and (possibly) react to anomalies. As with explainability in AI, it is possible that a human will immediately see why an anomaly has been labelled; it is also possible that the reason(s) will not be immediately apparent and in order to distinguish between anomalies and false alarms, reasons for the classification are required. An explanation can also assist in building confidence in the anomaly detection method,

and allow flaws to be detected where false positives are generated. Over-generation of false positives is a particular problem for anomaly detectors, and can rapidly lead to lack of trust in a deployed system, given that substantial work is often required to investigate an alarm.

The approach used in [8] finds outliers and anomalies; the authors use an "isolation forest" and "active anomaly discovery" (AAD) to improve identification of anomalies by incorporating user feedback. The isolation forest approach starts with a data set (numerical) and repeatedly selects a feature at random, then splits on values until there is only one point at each leaf. Regular data requires many more splits than outliers, hence anomalies are usually found at a shallow tree node. The process is repeated for (e.g.,) 100 trees, and the AAD system assigns anomaly scores and asks for user judgment on the highest ranked cases before re-calculating weights to downgrade outliers and upgrade anomalies according to the expert categorisation.

The work is extended in [21] to detect anomalies in event logs, by classifying each 30-min period (over 2 weeks) as anomalous or not, depending on the count of various events; by presenting the top 20 anomalous time intervals to an analyst, the scoring of anomalous events was refined in two stages and the number of false positives reduced (the authors state that this is due to feedback although it is possible that better results were due to looking at different sets within the data).

Saad et al. [22] consider the role of machine learning in malware detection, suggesting there is an over-optimistic view of AI as a solution to malware attacks. They point out that

> *"malware attacks in the wild continue to grow and manage to bypass malware detection systems powered by machine learning techniques. This is because it is difficult to operate and deploy machine learning for malware detection in a production environment or the performance in a production environment is disturbing (e.g., high false positives rate). In fact, there is a significant difference (a detection gap) between the accuracy of malware detection techniques in the literature and their accuracy in a production environment".*

The authors suggest use of if-then rules based on the input features used by a black-box classifier to approximate the malicious/benign decisions; such a representation is assumed to be adequate for communication with analysts.

Bolzoni et al. [23] indirectly address the issue of explainability by proposing a system that automatically classifies anomalies identified by an IDS into a taxonomy, thereby assisting analysts in determining the seriousness of the anomaly and possible actions that can be taken.

Explainability is also a key factor identified by Zhong et al. [24], who point out that analysts must be able to identify true positives in a large volume of noisy alerts from IDS and other sources, and rapidly decide whether the events constitute an attack, the nature of the attack, likely evolution of the attack, etc. Their proposed system aims to automate the initial filtering by learning a set of state machines from observation of analyst actions, so that knowledge is elicited and can be examined/transferred/reused.

The idea of using a framework based on human-understandable concepts is also introduced by [25], in the context of malware detection. In common with others, they point to the gap between laboratory studies and real-world deployment of machine-learning systems in cyber security and use a concept learning system based on description logic to develop tools to distinguish malware from benign binary files. This has the advantage that human expert knowledge can also be encoded and easily combined with the machine-generated knowledge. This can be interpreted as an anomaly detection task and provides a pointer to a more knowledge-based and understandable representation that could be useful in wider cyber-security anomaly detection.

Visualisation is also an important aspect of understanding how/why anomalies occur. The system described in [26] relies on external detection systems to find intrusions and gives insight into the communication patterns of the legitimate and suspicious traffic by means of high quality graphics—in this case, summarising NetFlow records on a large scale.

3. Routes to Explainability

Three primary approaches can be identified to explainable systems:

1. "Transparent-by-design" refers to systems that use a naturally understandable representation such as decision trees or rule-based classifiers. This rests on an assumption that symbolic representations are relatively easy to understand. The flaws in this assumption will be readily apparent to anyone who has tried to understand a convoluted SQL query, rule set, decision tree (or even a prolog program); however, for shallow trees and rule sets based on "natural" splits and groupings in the data, there is a good possibility that a human would be able to follow and/or highlight flaws in the reasoning process. For example, it should be relatively straightforward to understand why a medical diagnosis system produces a diagnosis of "common cold" by citing runny nose, sore throat, etc., as reasons for the diagnosis; one that explains its conclusion by producing a list of several hundred numerical variables and their allowed ranges would require considerably more insight and thought to make sense of. Case-based reasoning is another example of this category—it rests on an assumption that if a human can see why a decision has been reached in some typical cases, then they should be able to interpolate or extrapolate from those examples to a "similar" case.
2. Systems can be designed with explainability as a core component. This can be achieved by restricting the complexity of a machine learning model, or by building into the design features that support explanation. There are few (if any) software packages that follow this approach. Given that software libraries supporting black box models are relatively available and easy to use, writing bespoke software (or rewriting existing code) is generally a high cost route to explainability.
3. The most common approach is to use a black-box method in the first instance to develop a good classifier/predictor, and then adopt a "model-agnostic approach" [27] to build an explanation component as an add-on to the system. A post hoc explanation system fits one or more transparent-by-design components to each region of the input-output space [28]. For example, a set of decision trees could be built to replicate the input -output behaviour in restricted cases, with each tree handling a small part of the overall input space. Within each tree, it should be possible to see the key features and values that lead to a particular conclusion; hence it can be claimed that the system is interpretable. A similar interpretability case can be made for small rule sets, where the chosen features and attribute values indicate the important features, or a linear model where the important features can be identified by the weights.

We focus mainly on the third approach, implicitly including aspects of the first approach since "transparent-by-design" representations are the normal route to post hoc explanations.

Case-based reasoning can be a useful representation for post hoc explanation, since the specific cases chosen as exemplars should be well understood and, provided that the similarity metric is good, the parallel with a new case will make an obvious explanation for the decision. Keane [29] uses case-based reasoning as an add-on to a neural net classifier and provides a review of the coupling between case-based reasoning and neural nets (including deep learning systems), highlighting their origins in coupled neural net/case-based systems developed in the late 1990s.

It is, of course, possible to regard the black box model as a mapping from inputs to outputs, and attempt to understand (for an individual case) which of the inputs are most important in determining the output in that particular case. A popular approach, SHAP [30] uses Shapley values from game theory to assess the contribution of each input variable and hence to indicate which variables or combination of variables lead to each decision.

The use of post hoc interpretable models is popular in XAI literature as it is relatively easy to optimise the post hoc model against a metric (such as tree size) and to argue that the metric acts a proxy for interpretability. Although this makes research easier (and more reproducible) it is potentially misleading [31] as there is only intuitive support for a link between interpretability and tree compactness/explanation length/etc.

In general, the systems mentioned above assume the user has some expertise and understanding of the problem domain. Formalising this understanding by means of an ontology can feed into the explanation process—for example, in [32], decision trees are used to reproduce neural net results on two benchmark datasets and the authors demonstrate that (in these cases at least) decision trees refined with ontological knowledge are syntactically less complex and are better for human understanding than the unrefined versions. In this work, the ontology is used specifically to estimate the information content of a concept, and more general concepts are preferred as nodes in the trees.

We note also that the post hoc, individual case based approach can be useful for local explainability, i.e., understanding the input-output relation on a case-by-case basis, but is less suitable for global explainability, i.e., understanding the classification behaviour over the whole space The global approach is arguably more important for a human to understand when developing knowledge of the system behaviour (a cognitive model), and hence a degree of trust in the system's decisions. Lakkaraju et al. [33] argue that from this viewpoint, the idea of global model interpretability is more important than local interpretability and use so-called 2-level decision sets, in which the second level is a simple tree/rule set and the first level decides which second level to use.

In order for a human to develop a global understanding of a model, it is necessary to accurately predict outputs for any given input. As the internal workings of an algorithm become more complex, understanding the reasons for a particular decision becomes increasingly difficult and verifying that the decision is "correct" becomes impossible. At some point, it becomes easier to experiment with different input data than to understand the internal workings of a system.

Finally, a key point that is often neglected concerns the interface for explanation. In post hoc interpretability, the knowledge representation is fixed at design time, either as a decision tree, or a rule set, linear model, etc. However, the HCI community has identified richer and more expressive forms of interaction such as natural language (possibly restricted to some kind of formal query language), visualisation, and more specialised approaches such as symbolic mathematical models. There appears to be a great deal more "explanatory power" in an interactive process, where the user can request further detail or an alternative perspective on some aspects of the explanation. This is closer to a human-to-human mode of explanation but requires considerably more effort than today's simpler approaches. It represents an information transfer beyond the simple decision output, such as the input parameters that could be changed in order to change the decision, cause-and-effect links within the model, and general trust/confidence in the model output.

3.1. Explainability and Trust

The ultimate aim of explainability in a human–computer collaboration is to build trust in the machine's decisions and recommendations. It is clear that human teams function better when individuals trust each other's judgments and decisions, so we can expect a similar performance advantage when humans trust the automated agent(s) in a team. However, the notion of trust in an AI systems is difficult to define, quantify and measure—possibly even more difficult than explainability and interpretability—so we will leave this as an open question. It is interesting to speculate that trust could be expressed as a function of an AI system's accuracy and explainability (on the grounds that we trust a system if it gives correct answers and we understand why it has given those answers). In this case, it might be possible to estimate the degree to which a system is explainable from its accuracy and the degree of trust a user has in the system.

3.2. Evaluation

Assuming a satisfactory definition of explainability can be agreed, the obvious question for system evaluation and comparison is how to measure it (in cybersecurity or in the more general AI context).

There are many proposals, ranging from simple syntactic measures such as tree depth (for decision trees), through to complex experiments based on human factors and task efficiency. The former group is generally easy to quantify although it should be noted there is no direct relation between tree depth/rule length and understandability. There is an intuitive link between the two, in that a simple tree or rule set is likely to be more easily understood than a complex set; hence a proxy measure of tree/rule complexity can provide an indication but not a guarantee of interpretability. The second approach—evaluating interpretability by measuring overall system effectiveness—is much more difficult to set up and to repeat. It relies on a notion that if decisions are of better quality with the aid of the automated component, then the system must be understandable.

Expanding this view, Doshi-Velez and Kim [34] suggest that interpretability should be evaluated via one of three general approaches:

- Evaluating the system in its intended application, comparing outputs (decisions) with and without the explanation system to judge whether it improves overall performance or not. For example, a medical decision-making system could be evaluated by comparing patient outcomes.
- Evaluating the system on simplified applications that represent the real application in some way—for example, asking humans to judge the best explanation from several candidates.
- Using a proxy evaluation (such as tree depth), assuming that there is evidence linking the proxy to an application-based evaluation. In this way, it should be possible to avoid optimising the system without regard to real world performance.

In a study more focused on black box classifiers, Backhaus and Seiffert [35] proposed three criteria to compare interpretability:

- Ability to indicate important input features;
- Ability to provide typical data points representing a class;
- Existence of model parameters that indicate the location of decision boundaries.

The issue of interpretability can also be viewed in the light of moving beyond test-set accuracy as the primary measure of a classifier system, to considering its use in practical situations and developing metrics for usability. In particular (see [34]) there is little need for interpretability if there is no serious consequence of a wrong decision or if the system accurately reproduces the judgement of a human. For example, recognising numerals is a process that can be automated but the results are relatively easy for a human to validate.

In some senses the issue of judging interpretability of a system has many parallels to the Turing test and (as with the Turing test), a convenient approach is to turn the problem round. In the case of the Turing test, the question of how to define intelligence is replaced by a practical test that if a human judge cannot tell the difference between a computer and human response to arbitrary questioning, the computer must be classed as intelligent. In the case of interpretability, the equivalent argument states that if a system can satisfactorily explain its reasoning to a human then it is interpretable. This, of course, merely shifts the problem of evaluating a system's interpretability to evaluating the quality of an explanation.

The observation that explainability is a property of system and user, not just of the system, is fundamental to the analysis and survey presented in [36]. It is neatly summarised in the quote:

"explanations are social—they are a transfer of knowledge".

Nevertheless, the majority of studies that attempt to measure explainability fall back on looking at a simple proxy measure (such as tree complexity). In the area of neural nets coupled to case-based explainers surveyed by Keane [29], fewer than 5% carried out user-trials, a situation referred to as "the embarrassment of user testing" and highlighted as one of three main areas for future research.

3.3. History

Explainability is not a new problem for AI—the expert systems of the 1980's faced similar demands, even though they were largely based on symbolic representations and built by acquiring and codifying knowledge from human experts rather than from data. It seems that many current research projects linked to explainability are ignoring the experience and lessons available from this earlier work. Moore [37] is a frequently quoted review that surveys many of its contemporary systems and highlights the shortcomings of the simple "print out a trace of rules used" approach to explaining why a particular piece of data was requested or how a conclusion was reached.

As discussed in [38], three core issues governed the effectiveness of an expert system in real-world use—knowledge acquisition, knowledge representation, and the communication interface. Data driven systems are not concerned with the first aspect, since knowledge (or its equivalent) is extracted from data rather than from a domain expert; however, the remaining two components are highly relevant in explainability. Kidd observed that the knowledge representation must be able to capture the range and power of an expert's knowledge, and must be compatible at a cognitive level with the expert's view of the problem domain. In particular, simple rules were often found to be inadequate to express constraints, heuristics, causal links and interactions, and procedural knowledge. Furthermore, behaviour of rule-based systems is often implicitly dependent on internal factors such as the order in which rules are tried.

Regarding the communication interface, both Kidd and other workers (see for example, the discussion of mycin in [37]) noted the need for a flexible and two way "dialogue" so that exploration of "why" and "why not" questions could take place, and a full understanding of the reasoning process could be obtained.

This is not to say the explainability problem was solved at the time—typically, explanation was implemented in a simple manner, by presenting the rules and data used in deriving a conclusion. This approach that relied on the user understanding the representation and content of the knowledge base. A number of studies at the time highlighted the need for better interfaces and exchange between human and computer. For example, [39] reported that doctors using a medical expert system gave the highest rating to the "ability to explain diagnostic and treatment decisions", third highest to the ability to "display an understanding of their own medical knowledge", and rated "never make an incorrect diagnosis" as 14th out of 15 desirable properties.

The lesson for today's explainability researchers is that so-called understandable representations such as rules, decision trees, case-based reasoners, etc., may not be sufficiently expressive to convey the real reasoning underlying a data-driven decision-making process. Attention must also be paid to " cognitive aspects of the user interface, including dialogue control, explanation facilities, user models, natural language processing" [40]. The observation that

> *"Explainers must have alternative strategies for producing responses so that they may provide elaboration or clarification when users are not satisfied with the first explanation given. Furthermore, the system must be able to interpret follow-up questions about misunderstood explanations or requests for elaboration in the context of the dialogue that has already occurred, and not as independent questions"*

is just as relevant to today's AI systems as it was to the expert systems of the past.

4. Beyond Simple Representations

Again drawing on lessons from the past, Michalski [41], in an early paper on inductive learning (which can be regarded as a fore-runner of data-driven learning), commented that

> *"the results of computer induction should be symbolic descriptions of given entities, semantically and structurally similar to those a human expert might produce observing the same entities. Components of these descriptions should be comprehensible as single*

'chunks' of information, directly interpretable in natural language, and should relate quantitative and qualitative concepts in an integrated fashion"

This should be a guiding principle for explainable AI—the ability to express concepts in terms of simpler concepts, and to combine those concepts into higher level concepts as well as generating causal explanations. A concept encompasses a class of entities and constraints, properties, relations, etc. that distinguish the class of entities from other entities in the domain. This indicates the need for some kind of formal representation of a domain—not necessarily a full-blown ontology but at least a taxonomy of entities, and agreement on the scope and meaning of commonly used terms. Whilst not explicitly recommending use of an ontology, Weihs [42] introduced the ideas of mental fit and data fit, and highlighted the need to match concepts used in rules to those terms that make sense to the user.

A number of other studies highlight the requirement to define terms and concepts as a pre-requisite for exchanging knowledge between human and computer components of a system. The research reported in [43] starts from the common view that representations such as decision trees are inherently understandable, and that a decision tree (possibly using different splits in different areas of the input space) is a suitable "explanation" for a decision. They use decision trees to reproduce neural net results on two benchmark datasets and demonstrate that (in these cases at least) decision trees refined with ontological knowledge are less (syntactically) complex and, in some limited user-based evaluations, are better for human understanding than the unrefined versions. They use an ontology to define a generalisation/specialisation of concepts and prefer trees that have more general concepts. Based on trials they find these decision trees are more understandable than the unrefined versions. The understandability is assessed in three ways, by presenting examples to users (for manual classification using the tree), by presenting generic statements, what-ifs (what would you need to change in order to get a different outcome), and by pairwise tree comparison.

Riveiro and Thill [44] build on the idea that classification systems should use the same concepts as a user in explanations, suggesting additionally that "explanations should align with end user expectations"—in other words, when a user expects a particular output but the system produces a different output, the explanation should address why the expected output was not produced as well as why the actual output was produced. Importantly, they found their approach helped users to gain an understanding of the AI system by building an accurate mental model, providing additional evidence for the view that a shared understanding is a necessary component for effective explainable AI. This should be a global model, covering all cases, rather than a local model based on a few examples—for example, Chromik et al. [31] performed a set of experiments that illustrated how focusing on a few specific cases and simplified "local models" could be counter-productive, in leading users to a false or misleading idea of how the decisions are reached.

Chari et al. [45] provide a comprehensive historical overview of explainability in knowledge-based systems from the early 70s through the present day. They highlight the need for "Provenance-aware, personalized, and context-aware explanations", suggesting that the future lies in multi-actor systems which do not rely on the thought processes of a single individual (or the outputs of a single classifier system), but instead focus on the exchange and transformation of knowledge between multiple actors. The ability of the computer-based components to provide explanations, respond to queries and generally reference relevant information is a vital ingredient, and can only be implemented by a rich framework for knowledge representation within which humans and computer-based systems can interact Provenance of explanations is an important, and often neglected, issue. Information about the domain knowledge and sources of the data used in arriving at a decision is a key component in gaining user trust. In [45], provenance is proposed as a key component of explainability in knowledge systems, building on work in the semantic web using ontologies and knowledge graphs to structure and reason about explanations. The paper discusses taxonomies to classify explainable AI systems (using the IBM research framework) This includes the idea of an explanation taxonomy, which enables designers of

ML models to include features that will assist post hoc model interpretation. Note that other researchers such as Gilpin et al. [19] also propose taxonomies to categorise explanation capabilities, and Sokol and Flach [46] go further in providing multiple taxonomies for classification and comparison of different approaches under five distinct headings or "dimensions" (functional requirements, operational requirements, usability requirements, safety requirements, validation requirements). However, Chari et al., make a stronger case than most for the use of knowledge structures such as ontologies as part of the explanation process, rather than as a means of categorising or comparing the explanation process.

4.1. A Common Knowledge Framework

Considerable work has been devoted to systematically recording and structuring knowledge of cyberattacks. For example, the MITRE corporation has published (and continues to maintain) two resources which are widely used by the cyber-security community, namely CAPEC (capec.mitre.org, accessed on 1 November 2021) (Common Attack Pattern Enumeration and Classification), and ATT&CK (attack.mitre.org/, accessed on 1 November 2021) (Adversarial Tactics, Techniques, and Common Knowledge). These are closely related knowledge bases, with CAPEC focusing more on methods to exploit weaknesses in systems, whereas ATT&CK looks more at the lifecycle of an attack, such as how the vulnerabilities listed in CAPEC can be combined to achieve objectives. For example, exfiltration of data is an objective (listed in the ATT&CK database) that could be achieved via a specific attack such as SQL injection, exploiting a vulnerability in software (listed in the CAPEC database). ATT&CK also includes data on impacts, difficulty of attacks, possible paths to mitigate attacks, etc.

Both resources are structured in a way that makes them easily accessible to human experts (and to less-experienced readers); at the same time, the use of taxonomies and mark-up to structure the resources means they are relatively close to the format required for knowledge-based processing on a computer. They are primarily aimed at assisting human analysts, and hence contain much information in the form of natural language descriptions and summaries; extracting further information from the natural language text is a possibility, but not pursued here.

We briefly outline two possible routes to utilising the knowledge structure in existing resources (such as CAPEC and ATT&CK) in a collaborative cybersecurity system.

4.1.1. Using an Ontology (Knowledge Graph)

An ontology can be described as a conceptualisation of an application domain into a human understandable AND machine-readable format. It consists of a collection of concepts and their inter—relationships which collectively define an abstract view of that domain [47]. The purpose of an ontology is to define a shared vocabulary of terms for storing, exchanging and integrating knowledge. An ontology is designed to enable automated reasoning as well as to facilitate storage and validation of data. We regard knowledge graphs as an extension of ontologies, with more emphasis on graph operations/inference, and less on the semantics of the representation.

Much recent work has focused on the use of ontologies as part of the semantic web [48] and many of the tools developed for this purpose would be useful in developing a knowledge framework for a collaborative cybersecurity system.

Ontologies are designed for formal reasoning, e.g., using description logics or rule based approaches [49,50]. We can distinguish a spectrum of expressive and formal reasoning power, ranging from so-called lightweight ontologies (simply consisting of the concepts relations and properties) up to full ontologies which include axiomatic definitions and constraints. Inevitably a full ontology is more difficult to create and maintain. Reasoning processes are more sophisticated with a full ontology but are also typically more computationally intensive. In cases where the reasoning process is equivalent to first order predicate logic, it is possible to formulate undecidable queries—although most implementations based on description logics have efficient algorithms that are restricted in some way to

avoid such queries. Thus there are several reasons for preferring a simpler approach in the form of a lightweight ontology.

Typically, an ontology will enable reasoning about classes (e.g., checking subclass relations, or that classes are equivalent/disjoint), and reasoning about instances of a class.

The MITRE resources listed above are close to the format required for an ontology and (with good knowledge of XML/JSON transformations) can be converted into an ontology as described in [51], for example. We note that there is an implicit recognition of uncertainty in the original resources and hence in the derived ontology framework and instances—for example, attack impacts and the required skill levels are modelled by terms such as "very low" and "high likelihood". These are naturally fuzzy terms that have been approximated by precise categories.

In [51], a number of illustrative queries are posed, such as "which attack patterns have high impact, high likelihood of success and require high-level skills?" or "which malware is used by most threat actors". These queries are not easily answered by simply looking at individual items in the ATT&CK data but require consideration of whole classes.

Finally, we note that much of the human-accessible information is in (natural-language) description and comment sections, which could enhance the knowledge base after suitable processing. These descriptions can be used to clarify aspects of the formal ontology and to express more nuanced information than is allowed by the straightforward yes/no required by binary logic. For example, it is difficult to unambiguously distinguish between a severe impact and a very severe impact in the context of malware. As further examples, it might be "more likely than not" that a tool was developed by a nation-state, or "possibly true" that two pieces of malware were developed by the same actor.

4.1.2. Knowledge Graphs in Cybersecurity

A number of researchers have investigated the use of knowledge graphs for cybersecurity, typically starting from the Mitre databases. For example, Kurniawan [52] describes a system which integrates the ATT&CK and CAPEC data into an extended ontology, and then applies open source rules for log analysis to detect "indicators of compromise" in graphs derived from system logs. These indicators of compromise are linked back to the corresponding attack techniques, highlighting related concepts that could be relevant for analyst investigation. Whilst not explicitly addressing the interaction between human and machine, it is clear that this is intended as a collaborative tool using a representation that is both machine processable and human-understandable.

4.1.3. Using Formal Concept Analysis

Formal concept analysis (FCA) is a structured approach to analysing and classifying data that is stored in an "object—attribute" or "object—attribute—value" format. In this context, an object might be a specific piece of malware, or an actor engaged in attacks, and attributes could be severity of impact, skill level, etc. Formal concept analysis was originally developed for boolean-valued relations (i.e., an object either has an attribute or not) but can easily be extended to cases where there are multiple possible values (or even where the values are from a continuum, such as real numbers). Formal concept analysis yields a taxonomic structure, based on commonality of attributes, by grouping together all objects with specific attributes or supersets of these attributes. A concept is a set of objects, which cannot be distinguished on the basis of a specified subset of attribute values—in other words, the objects are indiscernible from each other when considering the specified attributes.

FCA takes a bottom-up approach to organising knowledge, based on the recorded objects and attributes, in contrast to development of an ontology which starts from human-defined classes.

Formal concept analysis works well in binary datasets (where an object either has or does not have a property) and where the data is not subject to noise and errors. Fuzzy extensions [53,54] model vague binary attributes. Non-binary attributes (values in a discrete

set or numerical range) are typically converted to multiple binary attributes. For example, a colour {red, green, blue} would be converted to three (mutually exclusive) binary attributes, colour = red, colour = blue, colour = green. Numerical attribute values are discretised and treated in the same way.

Figure 1 shows a fragment of the concept lattice extracted from the taxonomy in the Mitre ATT&CK dataset which categorises techniques into hierarchical classes. Figure 2 shows the same objects extracted from the CAPEC data, and grouped by the attributes: "likelihood_of _attack", "typical_severity" and "consequences" (such as "Access_Control", "Authentication", etc. It can be seen that the tactics are still closely linked, although the differing consequences associated with each measn thatthe hierarchical relationship is less clear. Note also that the "typical_severity" attribute is treated as though "High" and Very High" are completely separate categories. In practice there is a degree of subjectivity and overlap in the two labels, unless there is a clear and unambiguous definition that can delineate the two.

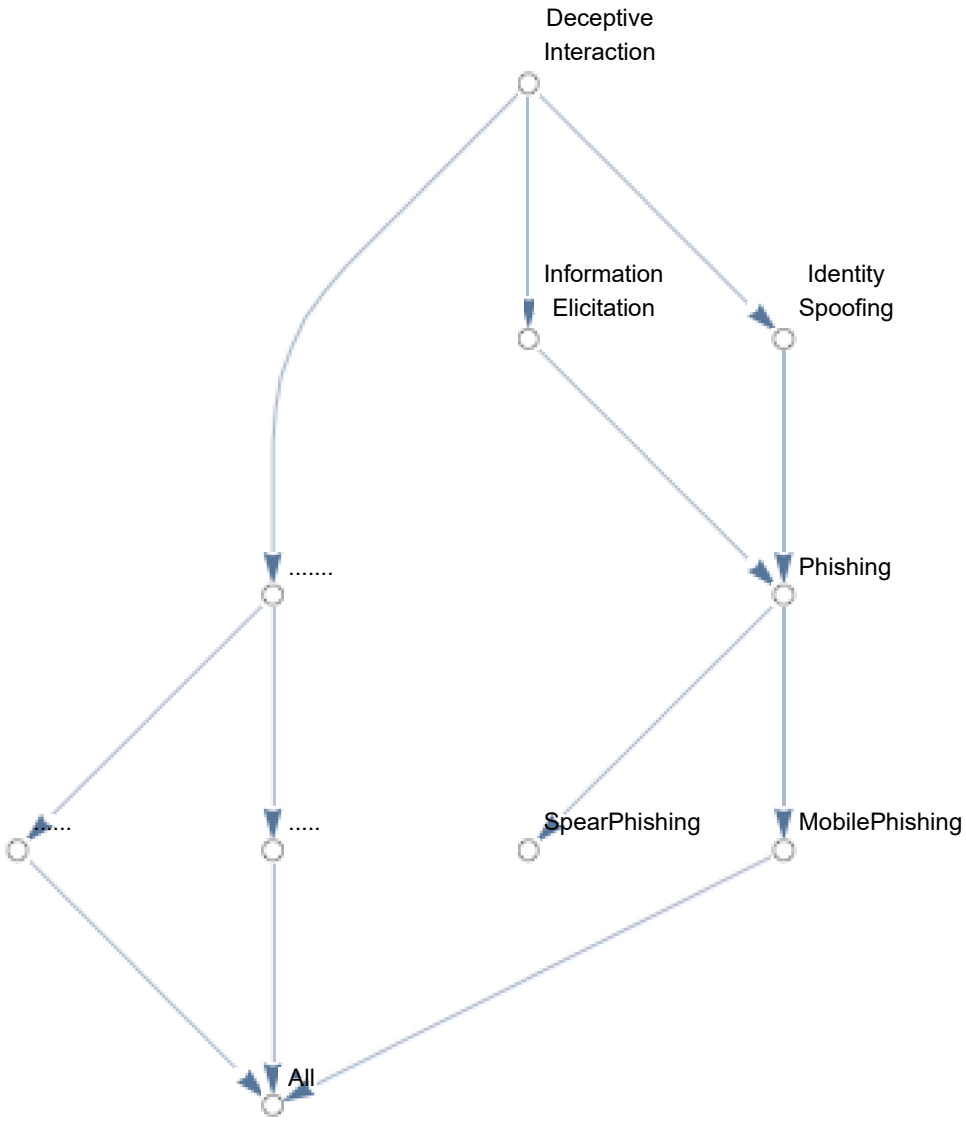


Figure 1. Fragment of a formal concept lattice for analysis of cyberattacks, taken from the ATT&CK datatset. The lattice is based on links between tactics, which are extracted from the dataset.

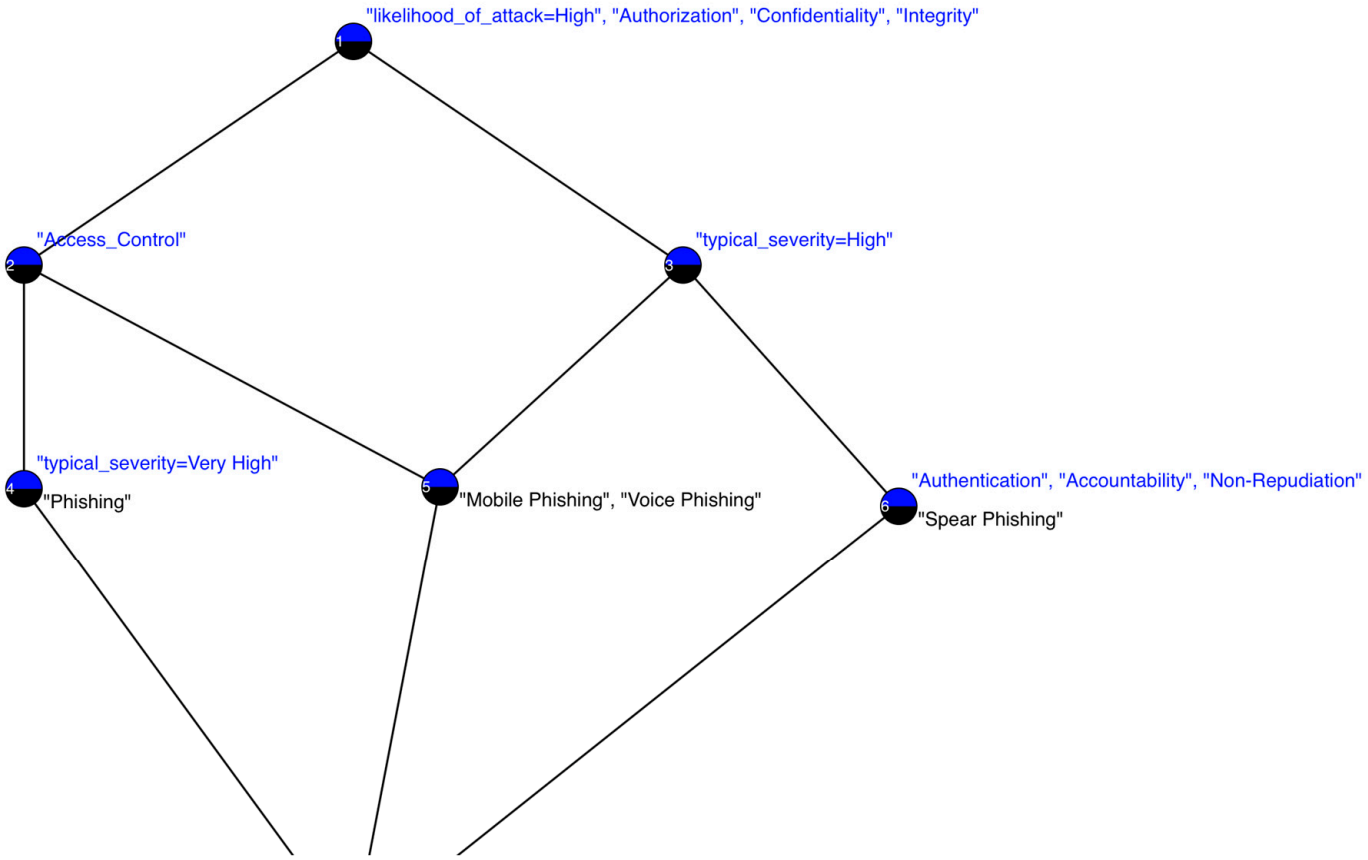


Figure 2. Fragment of a formal concept lattice for analysis of cyberattacks, taken from the CAPEC datatset and based on shared values for the attributes: "likelihood_of _attack", "typical_severity" and "consequences" (shown in blue). Objects (attack types) are shown in black.

If we choose to regard the category labels as fuzzy (in the sense that events may belong more or less strongly to a category), the approach described in [55] allows us to extend existing software to the case of fuzzy membership in categories, without needing to modify the underlying software.

4.2. Graded Knowledge Representation for Collaborative Intelligence

Machine processing is generally centered on well-defined entities and relations, ranging from the flat table structures of database systems through graph-based representations and up to ontological approaches involving formal logics. On the other hand, human language and communication is based on a degree of vagueness and ambiguity that leads to an efficient transmission of information between humans without the need for precise definition of every term used. Even quantities that can be measured precisely (height of a person or building, volume of a sound, amount of rainfall, colour of an object, etc.) are usually described in non-precise terms such as tall, loud, quite heavy, dark green, etc., More abstract properties such as beautiful landscape, delicious food, pleasant weather, clear documentation, corporate social responsibility, are essentially ill-defined, whether they are based on a holistic assessment or reduced to a combination of lower-level, measurable quantities.

Zadeh's initial formulation of fuzzy sets [56] was inspired primarily by the flexibility of definitions in natural language. He argued that most natural language terms (concepts) admit "graded membership", in that it is possible to compare two objects and to say whether or not one belongs more strongly to the concept. Clearly in the case of an elementary quantity such as height, we are generally able to say that *person-1* satisfies the concept "tall" better than *person-2* (or that they satisfy the concept equally well). Such gradation can be confirmed by measurement, if necessary. However, it is also valid to speak of graded

membership in the case of more complex concepts such as those listed above. We can generally rank the membership of different objects in the set representing the concept extensionin other words, the concept extension can be modelled as a fuzzy set. The interval [0, 1] is a convenient range for the membership function. It maps naturally to a scale where definite membership can be represented by 1, non-membership by 0, with intermediate values used to reflect the lack of a precise border between the two extremes. However, the fundamental requirement is that membership is ordered, not that precise memberships can be assigned on a scale.

In addition to the use of flexible terms, human reasoning is characterised by an ability to switch between different levels of granularity when dealing with a problem

More recent work [55,57] has combined the notions of formal concept analysis with graded tolerance relations. A concept is a set of objects, which cannot be distinguished on the basis of the describing attributes (intension). Given a partition of attribute values, equivalence classes naturally divide the objects into non-overlapping sets, each of which contains objects that are indiscernible on the basis of the attribute. For graded partitions, we obtain a nested sequence of lattices. We form a concept lattice using standard techniques—however, the lattice varies according to the membership grade used to create the partition.

5. Summary

The use of collaborative AI in cybersecurity requires seamless exchange of knowledge between agents, whether human or machine. Explainability is not a property of an intelligent system. It is a function of (at least) three elements—the task(s), the computer-based components in the process and the human participants. We should not spend time on philosophical discussions of explainability, but focus on whether or not the requisite tasks can be completed more effectively by the collaborative (computer + human) system when "explainability" is included in the collaborative interface. Graded formal concepts are a potential enabler for the exchange of information between humans and machines in a collaborative intelligent system, allowing us to model many of the soft definitions used in natural language descriptions of events.

Funding: This research was partly funded by BT under agreement ML853886.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Hawkins, D.M. *Identification of Outliers*; Springer: Cham, Switzerland, 1980. [CrossRef]
2. Barnett, V.; Lewis, T. Outliers in statistical data. In *Wiley Series in Probability and Mathematical Statistics Applied Probability and Statistics*; Wiley: Hoboken, NJ, USA, 1984.
3. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection for Discrete Sequences: A Survey. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 823–839. [CrossRef]
4. Denning, D.E. An Intrusion-Detection Model. *IEEE Trans. Softw. Eng.* **1987**, *13*, 222–232. [CrossRef]
5. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv.* **2009**, *41*, 15. [CrossRef]
6. Gupta, M.; Gao, J.; Aggarwal, C.C.; Han, J. Outlier Detection for Temporal Data: A Survey. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 2250–2267. [CrossRef]
7. Hoque, N.; Bhuyan, M.H.; Baishya, R.C.; Bhattacharyya, D.K.; Kalita, J.K. Network attacks: Taxonomy, tools and systems. *J. Netw. Comput. Appl.* **2014**, *40*, 307–324. [CrossRef]
8. Das, S.; Wong, W.-K.; Fern, A.; Dietterich, T.G.; Siddiqui, M.A. Incorporating Feedback into Tree-based Anomaly Detection. *arXiv* **2017**, arXiv:1708.09441.
9. Moustafa, N.; Keshk, M.; Choo, K.-K.R.; Lynar, T.; Camtepe, S.; Whitty, M. DAD: A Distributed Anomaly Detection system using ensemble one-class statistical learning in edge networks. *Futur. Gener. Comput. Syst.* **2021**, *118*, 240–251. [CrossRef]
10. Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *arXiv* **2019**, arXiv:1901.03407. Available online: https://ui.adsabs.harvard.edu/abs/2019arXiv190103407C (accessed on 1 January 2019).
11. Amarasinghe, K.; Kenney, K.; Manic, M. Toward Explainable Deep Neural Network Based Anomaly Detection. In Proceedings of the 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland, 4–6 July 2018.
12. Evangelou, M.; Adams, N.M. An anomaly detection framework for cyber-security data. *Comput. Secur.* **2020**, *97*, 101941. [CrossRef]

13. Duan, X.; Ying, S.; Yuan, W.; Cheng, H.; Yin, X. QLLog: A log anomaly detection method based on Q-learning algorithm. *Inf. Process. Manag.* **2021**, *58*, 102540. [CrossRef]
14. Yavanoglu, O.; Aydos, M. A review on cyber security datasets for machine learning algorithms. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017.
15. Sommer, R.; Paxson, V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010.
16. Castelvecchi, D. Can we open the black box of AI? *Nature* **2016**, *538*, 20–23. [CrossRef]
17. Gunning, D.; Aha, D. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Mag.* **2019**, *40*, 44–58.
18. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
19. Gilpin, L.; Bau, D.; Yuan, B.; Bajwa, A.; Specter, M.; Kagal, L. Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning. In Proceedings of the 5th IEEE International Conference on Data Science and Advanced Analytics, Turin, Italy, 1–3 October 2018.
20. Carvalho, D.V.; Pereira, E.M.; Cardoso, J.S. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* **2019**, *8*, 832. [CrossRef]
21. Siddiqui, A.; Stokes, J.W.; Seifert, C.; Argyle, E.; McCann, R.; Neil, J.; Carroll, J. Detecting Cyber Attacks Using Anomaly Detection with Explanations and Expert Feedback. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019.
22. Saad, S.; Briguglio, W.; Elmiligi, H. The Curious Case of Machine Learning In Malware Detection. In Proceedings of the International Conference on Information Systems Security and Privacy, Prague, Czech Republic, 23–25 February 2019; pp. 528–535. [CrossRef]
23. Bolzoni, D.; Etalle, S.; Hartel, P.H. Panacea: Automating Attack Classification for Anomaly-Based Network Intrusion Detection Systems. In Proceedings of the Recent Advances in Intrusion Detection, Saint-Malo, France, 23–25 September 2009; Springer: Berlin/Heidelberg, Germany, 2009. ISBN 978-3-642-04342-0.
24. Zhong, C.; Yen, J.; Liu, P.; Erbacher, R.F. Automate Cybersecurity Data Triage by Leveraging Human Analysts' Cognitive Process. In Proceedings of the 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), New York, NY, USA, 9–10 April 2016.
25. Svec, P.; Balogh, S.; Homola, M. Experimental Evaluation of Description Logic Concept Learning Algorithms for Static Malware Detection. In Proceedings of the ICISSP 2021, Online Conference, 11–13 February 2021; pp. 792–799. [CrossRef]
26. Fischer, F.; Mansmann, F.; Keim, D.A.; Pietzko, S.; Waldvogel, M. Large-Scale Network Monitoring for Visual Analysis of Attacks. In *Visualization for Computer Security*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 111–118. ISBN 978-3-540-85933-8.
27. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144. [CrossRef]
28. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. *arXiv* **2017**, arXiv:1705.07874.
29. Keane, M.T.; Kenny, E.M. How Case-Based Reasoning Explains Neural Networks: A Theoretical Analysis of XAI Using Post-Hoc Explanation-by-Example from a Survey of ANN-CBR Twin-Systems. In Proceedings of the Case-Based Reasoning Research and Development, Otzenhausen, Germany, 8–12 September 2019.
30. Antwarg, L.; Mindlin, M.R.; Shapira, B.; Rokach, L. Explaining Anomalies Detected by Autoencoders Using SHAP. *arXiv* **2019**, arXiv:1903.02407. Available online: https://ui.adsabs.harvard.edu/abs/2019arXiv190302407A (accessed on 1 March 2019).
31. Chromik, M.; Eiband, M.; Buchner, F.; Krüger, A.; Butz, A. I Think I Get Your Point, AI! The Illusion of Explanatory Depth in Explainable AI. In Proceedings of the 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, 14–17 April 2021.
32. Confalonieri, R.; Besold, T.R. Trepan Reloaded: A Knowledge-Driven Approach to Explaining Black-Box Models. In Proceedings of the ECAI, Santiago de Compostela, Spain, 29 August–8 September 2020; pp. 2457–2464. [CrossRef]
33. Lakkaraju, H.; Kamar, E.; Caruana, R.; Leskovec, J. Interpretable & Explorable Approximations of Black Box Models. *arXiv* **2017**, arXiv:1707.01154.
34. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, arXiv:1702.08608.
35. Backhaus, A.; Seiffert, U. Classification in high-dimensional spectral data: Accuracy vs. interpretability vs. model size. *Neurocomputing* **2014**, *131*, 15–22. [CrossRef]
36. Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* **2019**, *267*, 1–38. [CrossRef]
37. Moore, J.; Swartout, W. Explanation in Expert Systems: A survey. ISI Research Report RR-88-228. 1989. Available online: www.researchgate.net/publication/235125733_Explanation_in_Expert_Systemss_A_Survey (accessed on 7 July 2021).
38. Kidd, A.L.; Cooper, M.B. Man-machine interface issues in the construction and use of an expert system. *Int. J. Man-Mach. Stud.* **1985**, *22*, 91–102. [CrossRef]
39. Teach, R.L.; Shortliffe, E.H. An analysis of physician attitudes regarding computer-based clinical consultation systems. *Comput. Biomed. Res.* **1981**, *14*, 542–558. [CrossRef]

40. Berry, D.C.; Broadbent, D.E. Expert systems and the man? Machine interface. *Expert Syst.* **1986**, *3*, 228–231. [CrossRef]
41. Michalski, R.S. A theory and methodology of inductive learning. *Artif. Intell.* **1983**, *20*, 111–161. [CrossRef]
42. Weihs, C.; Sondhauss, U.M. Combining Mental Fit and Data Fit for Classification Rule Selection. In *Exploratory Data Analysis in Empirical Research*; Springer: Berlin/Heidelberg, Germany, 2003.
43. Confalonieri, R.; Weyde, T.; Besold, T.R.; Martín, F.M.D.P. Using ontologies to enhance human understandability of global post-hoc explanations of black-box models. *Artif. Intell.* **2021**, *296*, 103471. [CrossRef]
44. Riveiro, M.; Thill, S. "That's (not) the output I expected!" On the role of end user expectations in creating explanations of AI systems. *Artif. Intell.* **2021**, *298*, 103507. [CrossRef]
45. Chari, S.; Gruen, D.M.; Seneviratne, O.; McGuinness, D.L. Foundations of explainable knowledge-enabled systems. *arXiv* **2020**, arXiv:200307520.
46. Sokol, K.; Flach, P. Explainability fact sheets: A framework for systematic assessment of explainable approaches. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, 27–30 January 2020.
47. Gruber, T.R. A translation approach to portable ontology specifications. *Knowl. Acquis.* **1993**, *5*, 199. [CrossRef]
48. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Sci. Am.* **2001**, *284*, 28–37. [CrossRef]
49. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Said Tabet, S.; Grosof, B.; Dean, M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. 2003. Available online: http://www.daml.org/2003/11/swrl/ (accessed on 23 November 2021).
50. W3C. OWL Web Ontology Language Guide. 2003. Available online: http://www.w3.org/TR/owl-guide/ (accessed on 23 November 2021).
51. Grønberg, M. *An Ontology for Cyber Threat Intelligence*; Department of Informatics, University of Oslo: Oslo, Norway, 2019.
52. Kurniawan, K.; Ekelhart, A.; Kiesling, E. An ATT&CK-KG for Linking Cybersecurity Attacks to Adversary Tactics and Techniques. In Proceedings of the International Semantic Web Conference (ISWC), Online Conference, 24–28 October 2021; Available online: http://ceur-ws.org/Vol-2980/paper363.pdf (accessed on 23 November 2021).
53. Belohlavek, R.; Sklenar, V.; Zacpal, J. Crisply Generated Fuzzy Concepts. In Proceedings of the Third International Conference on Formal Concept Analysis, Lens, France, 14–18 February 2005.
54. Martin, T.P.; Abd Rahim, N.H.; Majidian, A. A General Approach to the Measurement of Change in Fuzzy Concept Lattices. *Soft Comput.* **2013**, *17*, 2223–2234. [CrossRef]
55. Martin, T.P.; Azvine, B. Graded Concepts for Collaborative Intelligence. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018.
56. Zadeh, L.A. Fuzzy Sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
57. Martin, T.P.; Azvine, B. Graded associations in situation awareness. In Proceedings of the 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), Otsu, Japan, 27–30 June 2017.