

Self-Aware Effective Identification and Response to Viral Cyber Threats

Pietro Baroni

University of Brescia
Brescia, BS, Italy

Daniela Fogli

University of Brescia
Brescia, BS, Italy

Massimiliano Giacomin

University of Brescia
Brescia, BS, Italy

Giovanni Guida

University of Brescia
Brescia, BS, Italy

Federico Cerutti

University of Brescia
Brescia, BS, Italy
and
Cardiff University
Cardiff, Wales, United Kingdom

Francesco Gringoli

University of Brescia
Brescia, BS, Italy

Paul Sullivan

Intelpoint Inc.
Springfield, Virginia, United States

Abstract: Artificial intelligence (AI) techniques can significantly improve cyber security operations if tasks and responsibilities are effectively shared between human and machine. AI techniques excel in some situational understanding tasks; for instance, classifying intrusions. However, existing AI systems are often overconfident in their classification: this reduces the trust of human analysts. Furthermore, sophisticated intrusions span across long time periods to reduce their footprint, and each decision to respond to a (suspected) attack can have unintended side effects.

In this position paper we show how advanced AI systems handling uncertainty and encompassing expert knowledge can lessen the burden on human analysts. In detail:

- (1) Effective interaction with the analyst is a key issue for the success of an intelligence support system. This involves two issues: a clear and unambiguous system-analyst communication, only possible if both share the same domain ontology and conceptual framework, and effective interaction,

- allowing the analyst to query the system for justifications of the reasoning path followed and the results obtained.
- (2) Uncertainty-aware machine learning and reasoning is an effective method for anomaly detection, which can provide human operators with alternative interpretations of data with an accurate assessment of their confidence. This can contribute to reducing misunderstandings and building trust.
 - (3) An event-processing algorithm including both a neural and a symbolic layer can help identify attacks spanning long intervals of time, that would remain undetected via a pure neural approach.
 - (4) Such a symbolic layer is crucial for the human operator to estimate the appropriateness of possible responses to a suspected attack by considering both the probability that an attack is actually occurring and the impact (intended and unintended) of a given response.

Keywords: *cyber threat intelligence, machine learning, artificial intelligence*

1. INTRODUCTION

The evolution of digital-enabled activities in recent years, also boosted by the COVID-19 pandemic, led to profound changes across the digital value-chain, where new challenges have emerged and have significantly affected the cyber security industry. Cyber security risks are and will become harder and harder to assess and interpret due to the growing complexity of the threat landscape, the adversarial ecosystem, and the expansion of the attack surface [1]. This will boost the spread of attacks from Advanced Persistent Threats (APTs) [2], where fleets of sophisticated attackers constantly try to gain and maintain access to networks and the confidential information that is contained within them, or to use them as a starting point for further attacks.

To illustrate the complexity of attacks from APTs, let us refer to the Lockheed-Martin *cyber kill chain* model [3], which distinguishes seven phases attackers usually follow:

1. *Reconnaissance*: Research is conducted to identify the targets appropriate to meet planned objectives.
2. *Weaponization*: Malware is coupled with an exploit into a deliverable payload.
3. *Delivery*: Malware is delivered to the target.
4. *Exploitation*: A vulnerability is exploited to gain access to the target.

5. *Installation*: A persistent backdoor is installed on the victim’s system to maintain access over an extended period of time.
6. *Command and Control (C2)*: Malware establishes a channel to control and manipulate the victim’s system.
7. *Actions on objectives*: After progressing through the first six phases, which might take months, the intruder, having access to the victim’s system, can easily accomplish the mission goals.

The cyber kill chain model [3] also illustrates the defence options, namely: *detect*, *deny*, *disrupt* (e.g. in-line antivirus), *degrade* (e.g. throttling the communication), *deceive* (e.g. using decoys such as honeypots), and *destroy*.

While [3] does not provide specific guidance on choosing between the various options, [4, Fig. 7.1] illustrates how defence – specifically for APTs – is an iterative process comprising three steps: *sense* (continuously sensing adversary actions), *observe* (continuously estimating intent and the capabilities of the adversary), and *manipulate* (delivering cyber deception based on observations). In this paper, we expand on the second step, the estimation of the intent and capabilities of adversaries, and embed this into the cyber threat intelligence framework (Section 2).

When focusing on cyber threat analysis, the amount of data that needs to be processed, the tempo, and the inevitable presence of adversarial actors assembles unique challenges that require advanced artificial intelligence (AI) capabilities. Our main contribution lies in Section 2, where we illustrate the desiderata for the effective usage of AI capabilities in cyber threat analysis. In the rest of the paper, we also discuss possible – albeit not all – techniques and technologies to satisfy such desiderata, most of which are based on previous work some of us directly contributed to. Our focus is on APT attacks that necessarily require a human analyst to assess the situation: less dangerous threats can be mitigated with existing tools and techniques, and this will not be part of our investigation. In Section 3 we discuss in detail the role of the human analyst,¹ who is pivotal for the success of cyber threat intelligence. Furthermore, systems need to be aware of the presence of adversarial and deceptive actors, hence in Section 4 we discuss the need for accurate quantification of uncertainty and propose a preliminary approach to this problem for raw data. In Section 5 we discuss the need to identify complex activities linked by temporal and causal relationships. Finally, in Section 6 we focus on the strategic thinking involved in choosing between alternative courses of action to manage APT attacks, in particular that which concerns unwanted side effects that might enable the attackers to acquire information of the analyst’s state of knowledge and intentions, making them aware that she is aware of their attack.

¹ To avoid the awkwardness of strings of *he or she*, we borrow a convention from linguistics and consistently refer to a generic intelligence analyst of one sex and a generic decision-maker of the other. The female gender won the coin toss, and will represent the intelligence analyst. Attackers will always be referred to in the plural.

2. BACKGROUND AND DESIDERATA

Cyber threat intelligence is a cyclic process that analysts use to produce knowledge about weaknesses in one or more assets in an organisation that can be exploited by one or more threats. Like traditional intelligence analysis [6], [5], it comprises several steps which, merging the contributions from some of the seminal works in the field,² can be summarised as follows:

1. *Direction setting*: The decision-maker poses a question or requests advice (intelligence requirement): we assume this step consists in identifying APT attacks and the side effects of countermeasures.
2. *Data collection*: The analyst collects raw data – network logs from the firewalls of her organisation – into shoeboxes.
3. *Data collation*: She imposes a standard format – standardising the attributes for each log entry – to the data in the shoeboxes to create an evidence file.
4. *Data processing*: She injects useful semantics (for her task), or *schema*, in the data; for instance, by searching for classes of information such as downloads of malware.
5. *Data analysis*: She creates a case for or against the detection of APT attacks by leveraging causal links from within the data, thus building reasonable hypotheses. If under attack, she estimates the intent and capabilities of the attackers, and highlights issues with available courses of action.
6. *Dissemination*: She identifies the relevant pieces of knowledge for the decision-maker and prepares a presentation that needs to be disseminated to the decision-maker.
7. *Feedback*: She reacts to feedback from the decision-maker, who might ask for explanations or relevant details left out of the report, and that might become a new intelligence requirement.

Three main loops are identified over these steps [6]: the policy loop, which corresponds to the process leading to the identification of intelligence requirements; the foraging loop, which moves data from sources to evidence files; and the sensemaking loop, which processes data into information and knowledge shared with the decision-maker.

In this paper we focus on the first two activities associated with the sensemaking loop, namely data processing and data analysis. While dissemination and feedback are also vital for the success of the enterprise, we will not discuss these in this paper, thus silently dropping the decision-maker from the frame. An interested reader is referred to [8] and [7] for discussions on how AI can help with writing intelligence reports.

² Prunckun [5] merges dissemination and feedback, while Pace *et al.* [46] do not distinguish between information collection and data collation, and between report writing and dissemination. Prunckun [5] also names step 2 as “Information collection” following the data to wisdom hierarchy [47].

Within the scope of our study, we introduce four desiderata that AI systems need to satisfy to effectively support the analyst.

D1: Putting the analyst at the centre. While the analyst is a highly educated and skilled professional, intelligence support tools should minimise the risk of misunderstandings and allow for frequent interactions. Studies on the quality of the interaction with AI-based systems for situational understanding are scarce, especially in the cyber security domain. The analyst should be involved from the very first steps of a project and participate in all design decisions. In particular, the interface between systems and the analyst should be based on a shared ontology, familiar to the analyst but at the same time precise enough to be used by an automated system. The ontology needs to be co-designed and continuously refined on the basis of analyst feedback. Moreover, systems should allow the analyst to ask for justifications of the reasoning path followed by the systems and underlying the obtained results. The possibility to ask questions (and receive appropriate answers) contributes by providing her with the feeling of having investigated all relevant issues and checked system reasoning. In this way, the “not invented here” syndrome [9] can be avoided and trust in system advice and acceptance can significantly increase.

D2: Embracing uncertainty. There is no such thing as a perfectly certain datum in the real world: everything comes with shades of uncertainty. Traditional uncertainty estimation approaches in AI aim at quantifying it via probabilities and this can be highly misleading. Indeed, there are (at least) two different sources of uncertainty, namely aleatoric and epistemic uncertainty [10]. *Aleatoric uncertainty* refers to the variability in the outcome of an experiment which is due to inherently random effects (e.g. flipping a fair coin): no additional source of information but Laplace’s daemon [11, p. 4] can reduce such variability. *Epistemic uncertainty*, instead, refers to the epistemic state of an agent; hence, it is determined by a lack of knowledge that, in principle, can be reduced on the basis of additional data.

For instance, we can create a vanilla neural network with a softmax final layer that takes as input a dataset of Portable Executable (PE) headers⁴ of pieces of software labelled either *normal* or *malware*,⁵ and, for any new PE header, it returns an assessment of it being *normal* or *malware*. Figure 1 illustrates the case⁶ with purple dots representing normal software and blue representing malware in the considered dataset. The yellow area represents the confidence in the class prediction: the darker the yellow, the lower the confidence. The dark yellow area lies on the class boundary

³ That is, a negative attitude to knowledge that originates from a source outside the own institution.

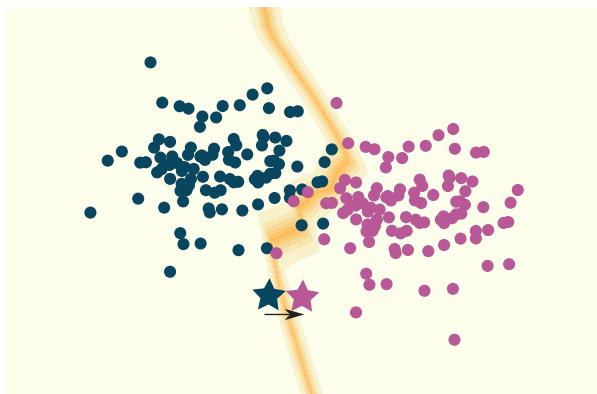
⁴ Portable Executable (PE) is the format in which Microsoft Windows requires executables to be encoded. It is composed of headers and various data and code sections. For further details, <https://docs.microsoft.com/en-us/windows/win32/debug/pe-format#overview> (accessed 5 Dec 2020).

⁵ We limit ourselves to two classes only for illustration purpose.

⁶ Clearly Figure 1 does not represent a real dataset: here we simplified it substantially by generating a toy 2D dataset for clarity of presentation.

and is a manifestation of aleatoric uncertainty: pieces of software close to such a boundary have characteristics so similar that distinguishing between them is hard, and no additional data can change this. Instead, the lighter yellow areas represent regions of high confidence, and that can be the case despite the fact that no data is present there. This is a by-product of using the commonly adopted softmax approach that divides the entire space of parameters into the set of classes it is trained for (closed world), thus leaving no room for uncertainty. This is also the main characteristic exploited in adversarial machine learning (like the Fast Gradient Sign Method [12]), where a data sample can be modified imperceptibly for a human but enough for a misclassification. Consider, for example, the blue star in Figure 1, a piece of malware that was not part of our original dataset and is close to the class boundary. A very limited modification of its attributes could transform it into the purple star on its right, which would then *magically* transform it into normal software with high confidence. A clear, honest assessment of the reliability of predictions is necessary.

FIGURE 1: NORMAL SOFTWARE (PURPLE) VS MALWARE (BLUE) CLASSIFICATION WITH CONFIDENCE LEVEL USING SOFTMAX: THE BRIGHTER THE AREA (OF YELLOW), THE GREATER THE CONFIDENCE



D3: Recognising complex events. An APT attack is a chain of events linked together by time and causality [3]: it is the result of a deliberate design led by human attackers. AI systems need to be equipped to reason not only about the detection of single events but also, and more importantly, to recognise events linked together by time and causality (i.e. complex events) [13]. They also need to easily adapt to evolving environments, where changes can occur in very rapid or very slow time frames. Having a perfect immutable detector of APT attacks at each stage based on past knowledge gives the analyst very little advantage in a world where new vulnerabilities are discovered each minute.

D4: Strategic thinking. When the analyst estimates the intent and capabilities of attackers, she must highlight potential side effects of the available courses of actions [3], namely: *do nothing* (always an option), *deny* (often the most common), *disrupt*, *degrade*, and *deceive*.⁷ For the decision-maker, to choose rationally among these, the value of the information the attackers could acquire from the effects of the chosen countermeasures should be carefully pondered. It would indeed be naïve to assume that the attackers are not operating their own intelligence process. The analyst needs to consider the risk that the attackers might discover that she has some level of awareness of their operations (see the concept of *high-order theory of mind* [14]). There might also be cases where *do nothing* is a reasonable choice, like in the case of the accident at the Lawrence Berkeley Laboratory (USA) in August 1986 [15], where persistent intruders were found in a relatively low-value network as part of an “island-hopping” attack [16] towards a much higher value target. By tracing their activities for nearly one year, and employing deception warily, the attackers were found and proved to be connected to the KGB [17].

In the following sections we expand on technical solutions that can help satisfy each of the four desiderata illustrated above.

3. PUTTING THE ANALYST AT THE CENTRE

Supporting the analyst’s critical thinking when facing complex, intricate menaces from APTs is not trivial. To benefit from using decision-support AI systems, the analyst must have an appropriately calibrated level of trust in the system [18], [19]. Trust is well calibrated when she sets her trust level appropriately to the AI’s capabilities, accepting the output of a competent system but employing other resources or her own expertise to compensate for possible AI errors; conversely, poorly calibrated trust reduces team performance because she might trust erroneous AI outputs or not accept accurate ones [18], [20], [19].

Two problems stand out. First, it is necessary to create a stock of shared knowledge between the analyst and the artificial system she is using in order to understand the complex assessments that generally come with data analysis through machine learning and the intricate relationships between events composing an attack. Second, the analyst needs to be allowed to question the system and receive justifications for the results obtained and the reasoning processes behind them.

As far as the former issue is concerned, we argue in favour of using shared ontologies – as part of the community has already started doing; for example, [21] where domain entities and the relationships between them can be explicitly represented, thus

⁷ Albeit *destroy* is also an option, we will not investigate it in this paper.

clarifying the semantics for each of them. In this way, it is easier to share concepts with AI tools, as well as collecting and representing the analyst's knowledge and experience. However, querying such ontologies, thus allowing her to navigate the inevitable intricate, interrelated structures in it, soon becomes very challenging. Visual inspection is ineffective beyond a certain size threshold, while existing query languages, such as SPARQL, are often beyond the abilities of an analyst. We argue in favour of Controlled Natural Languages (CNL); in other words, “engineered subsets of natural languages whose grammar and vocabulary have been restricted in a systematic way in order to reduce both ambiguity and complexity of full natural languages” [22], while not being so restricted as a formal language. In particular, highly precise and expressive CNLs – according to the classification provided in [23] – could potentially be used as an intermediate representation between analyst and AI system, and some have also been employed in preliminary studies to facilitate human-machine joint analytical processing [24], [25], although a comprehensive assessment is still lacking. Last but not least, ontologies should also account for the uncertainty that inevitably affects all steps in the cyber threat intelligence cycle: a clear representation and communication of uncertainty plays a central role in building trust [26], [19].

As far as the latter issue is concerned, the possibility to ask the system for and obtain detailed justifications about the advice provided and the reasoning path exploited for its generation is of paramount importance. We argue that interactive interfaces must support an effective analyst-driven dialogue with the AI system, either at specific steps in the intelligence process or according to an interrupt-based protocol, where the user is allowed to ask the system at any moment during its operation. The dialogue can be based on CNL and organised according to a simple question-answer schema we co-designed [25], allowing, however, for a variety of questions that cover most of the possible information needs of the analyst, such as, for example, “justify your advice” (make the reasoning process behind the advice explicit) or “show alternatives” (illustrate possible alternative analyses and explain why they have been discarded).

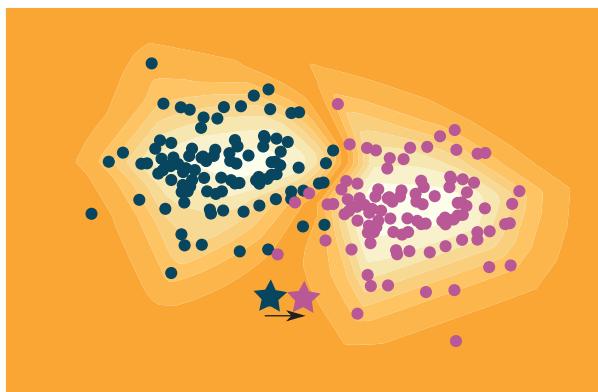
4. EMBRACING UNCERTAINTY

Uncertainty pervades the entire cyber threat intelligence cycle, including the recognition of complex events linked by temporal and causal relations (Section 5) and strategic reasoning about the side effects of possible countermeasures (Section 6). Due to space constraints in this paper, we focus only on uncertainty when processing data.

The Bayesian paradigm of mathematical statistics is one of the most powerful tools we have for estimating aleatoric and epistemic uncertainty. It is based on an

interpretation of probability as a rational, conditional measure of uncertainty [27]. Pure Bayesian methods are unfeasible due to the gargantuan amount of data needed, but they can be approximated by using, for instance, Evidential Deep Learning with Noise Contrasting Estimation (EDL-NCE), which we co-designed [28], [19] and that requires less computational power and data. The main approximation behind EDL-NCE is that the posterior probability that, for instance, recalling our example from Section 2, a piece of software \vec{x} is malware, $p(\text{malware} | \vec{x})$, is forced to be Beta-distributed or Dirichlet-distributed if we are considering more than two classes.⁸ Figure 2 illustrates the result of the classification using EDL-NCE [28] on the dataset we introduced in Section 2 (see Figure 1). From a visual inspection we can appreciate how EDL-NCE derives an implicit class density estimation represented by the shades of yellow illustrating the confidence in the classification in Figure 2.

FIGURE 2: NORMAL SOFTWARE (PURPLE) VS MALWARE (BLUE) CLASSIFICATION WITH CONFIDENCE LEVEL USING EDL-NCE [28]: THE BRIGHTER THE AREA, THE GREATER THE CONFIDENCE



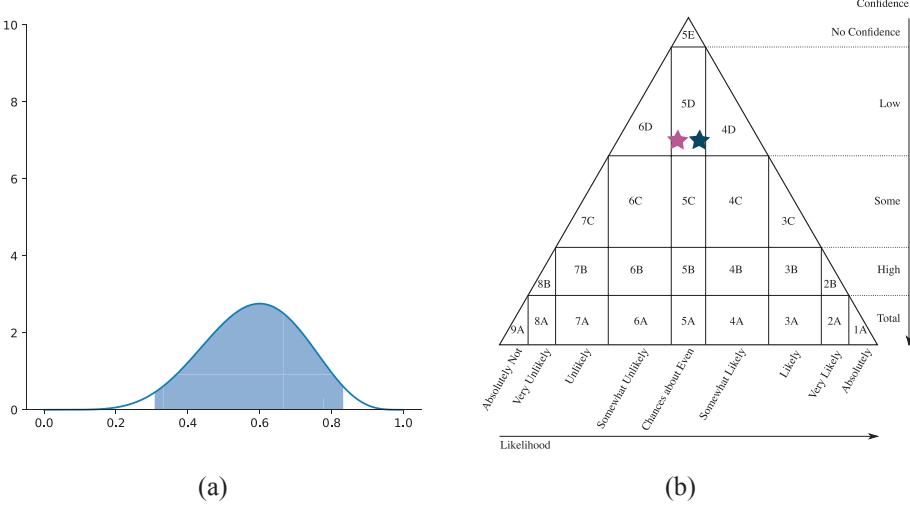
A Beta distribution needs two parameters representing the amount of evidence in favour of the two classes we are considering. For instance, let us consider again the star datapoint in Figure 2: by using EDL-NCE, we can compute $p(\text{malware} | \vec{x}) = \text{Beta}(7,5)$ (Figure 3(a)), which informs us that we have slightly more evidence in favour of it being a piece of malware than the opposite. With reference to Figure 3(a), we can note that (1) the expected value is 0.583, thus suggesting we are very close to the class boundary and then we have high aleatoric uncertainty, and (2) the variance

⁸ Random variables with two outcomes (e.g. tossing a coin, or detecting normal software vs malware) are known to follow the Bernoulli distribution $f(\vec{y} | \pi) = \pi^{\sum y_i} (1-\pi)^{n-\sum y_i}$ (for $y \geq 1$). If we then want to assess the value of π from some given data samples, we can use the Bayes theorem to compute the *posterior distribution* $g(\pi | \vec{y}) \propto g(\pi) f(\vec{y} | \pi)$ on the basis of a chosen prior $g(\pi)$. Since we know that Beta distribution is the conjugate for the Bernoulli, given $g(\pi) = \text{Beta}(\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \pi^{\alpha-1} (1-\pi)^{\beta-1}$ (for $0 \leq \pi \leq 1$), we have that $g(\pi | \vec{y}) \sim \text{Beta}(\alpha + \sum y_i, \beta + n - \sum y_i)$ that represents a distribution of probabilities for the phenomenon we were addressing. The generalisation of the Beta distribution to $k > 2$ outcomes (e.g. rolling a dice) is the Dirichlet distribution.

is 2.07E-2, thus suggesting a rather wide 95% confidence interval – identified by the shaded blue area under the curve of the distribution in Figure 3(a) – and then we have rather high epistemic uncertainty. Resuming the discussion presented in Section 2 (see point D2), note that a tiny manipulation of the PE headers of the star datapoint considered above might transform it into a Beta distribution with an expected value of less than 0.5; therefore, using a discriminative approach like softmax, it would be classified as normal software, but it would not have much effect on the epistemic uncertainty, and thus on the 95% confidence interval.

Beta distributions can be mapped directly into subjective logic opinions [29], namely a tuple of three values representing the *belief*, *disbelief*, and *uncertainty* in a given proposition. Since the three values must be non-negative, and must sum up to one, they identify a triangle in a 3D space that can easily be flattened in the 2D space depicted in Figure 3(b) as each subjective logic opinion becomes a point in the triangle [30]. In it, the vertical is the axis of confidence and it is a direct manifestation of epistemic uncertainty, from *no confidence* to *total confidence*; while the horizontal is the axis of likelihood, linked to aleatoric uncertainty, from *absolutely not likely* to *absolutely likely*. This space can be divided into different regions, each of which can be associated by a code – for example, 4C, similar to the admiralty code [5] already in use in the intelligence community – and by a couple of textual labels, such as *somewhat likely* with *some confidence*. Thanks to our previous evaluation of interfaces for decision support exposing labels representing subjective logic opinions, we argue that this has potential for creating understanding about epistemic and aleatoric uncertainty in highly-skilled personnel [25], which an analyst is supposed to be, and anecdotal evidence also suggests that decision-makers, such as physicians, appreciate the possibility of rapidly comparing the uncertainty associated with multiple reports using visual inspections of areas within the triangle. In our example, from Figure 3(b), we can see that the very same tiny manipulation that would have led a softmax approach to misclassify malware as normal software with high confidence now would not have much effect: in both cases – the original and the manipulated one – the classification shows that *chances are about even with low confidence*.

FIGURE 3: (A) GRAPHICAL REPRESENTATION OF $p(\text{malware} | \vec{x}) = \text{Beta}(7,5)$, FOR \vec{x} BEING THE STAR DATA SAMPLE IN FIGURE 2. (B) IN BLUE THE REPRESENTATION OF $\text{BETA}(7, 5)$ AS A SUBJECTIVE LOGIC OPINION AND IN PURPLE THE REPRESENTATION OF THE CLASSIFICATION OF THE MANIPULATED INPUT INTO THE PURPLE STAR IN AN ELABORATION OF JØSANG'S [29, P. 49] SPACE OF SUBJECTIVE LOGIC OPINIONS



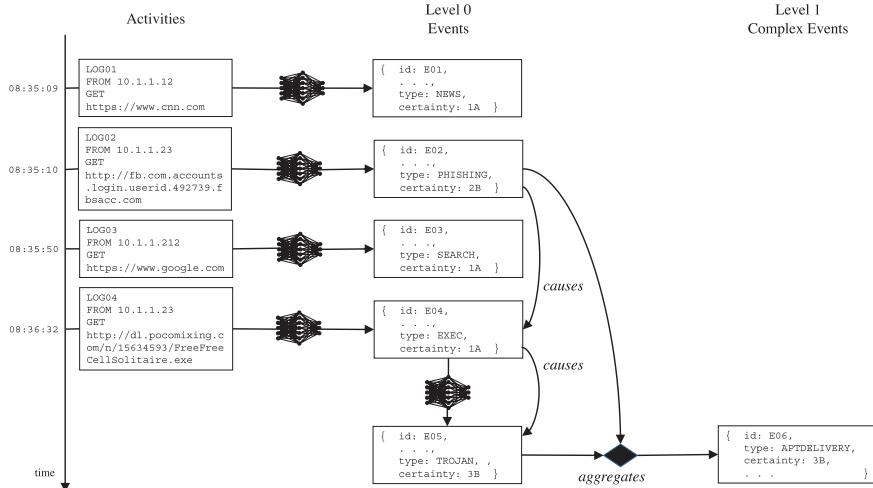
An approach to uncertainty based on subjective opinions can yield much more robust systems and, as discussed in Section 3, can help build trust with the analyst.

5. RECOGNISING COMPLEX EVENTS

Each APT attack is a set of chains of events linked together by time and causality [3] (see Section 1). We therefore advocate the use of complex event processing systems [13]. Following Luckham's [13] definitions – and differently from the everyday usage of the term event – an event is a computing object that signifies an activity that has happened. It has attributes such as the activity it represents and a timestamp or time intervals. Events can be linked by relationships of time, causality, and aggregation. If event A represents an activity that consists of the activities signified by a set of events B_1, B_2, \dots, B_n , then A is a complex event; in other words, it is an aggregation of all the events B_i ; conversely, B_i are members of A . Aggregation is a tool for making the activities in a complex system understandable to humans [13] and is the fundamental component in an *event abstraction hierarchy* that induces a sequence of levels such that the events in each level are defined on the basis of an aggregation of events at previous levels via aggregation rules. Clearly this applies to all the levels except the first one – conventionally Level 0 – which does not contain complex events.

For instance, the analyst might start looking into the evidence file containing network traces of HTTP(S) connections. Figure 4 illustrates the situation we specifically devised for this research: on the left there are the network logs collected in the evidence file. Activities are transformed into events thanks to a neural network trained to detect the type of URL, which might be a NEWS outlet, or a SEARCH engine, or the beginning of a download – an EXEC file, or even a PHISHING website (i.e. cloning a website to pose as it while delivering a malicious payload). Events might trigger the creation of other events: in the figure, downloading an EXEC has led to analysing it using a malware detector similar to the one we illustrated in Section 4, and that concluded that it is *likely with high confidence* (certainty: 3B) that it is a trojan; that is, malware misleading the user about its intent. An aggregation rule is then triggered and generates a complex event that signifies the detection of the delivery of a weapon as part of possible APT attacks.

FIGURE 4: ILLUSTRATION OF DETECTING THE DELIVERY OF AN APT WEAPON AS A COMPLEX EVENT. TIME FLOWS FROM TOP TO BOTTOM. ARROWS MARKED WITH  REPRESENT DATA PROCESSING VIA NEURAL NETWORKS. ATTRIBUTES ARE REPRESENTED IN JSON-LIKE SYNTAX. ARROWS LABELLED WITH *CAUSES* REPRESENT CAUSAL RELATIONSHIPS BETWEEN EVENTS. BLACK DIAMONDS REPRESENT AGGREGATION RULES.



Two limiting factors emerge. The first is the identification of relationships between events, in particular the aggregation rules. They can either be elicited by domain experts, or they can be learnt from annotated datasets of sequences of events by leveraging, for instance, inductive logic programming [31]. However, it is beyond doubt that high-quality [32] domain knowledge expressed as aggregation rules must

be curated and maintained, and this adds additional weight to the usage of suitable (controlled natural) languages (see Section 3).

The second problem is linked to adaptability to new weapons. Existing approaches using complex event processing for detecting APT attacks (e.g. [34] and [33]) assume that each of the models used to create events is separately trained on top of existing curated datasets: in the world of viral threats, this lack of adaptability is unsustainable. Adaptability to new contexts is the basis of novel, neuro-symbolic approaches to (simplified) complex event processing [35], [36]. Such approaches, which we co-designed, require as input only raw pieces of information (the logs identifying the activities, left of Figure 4), sets of aggregation rules, and the final labels. By leveraging approaches such as [37], linking together symbolic knowledge (aggregation rules) with sub-symbolic data processing (the identification of events from raw data), they can train classifiers for the various events of interest, such as PHISHING, EXEC and TROJAN, without the need to provide specific information about them. Using synthetically generated raw data, we gathered evidence in favour of this, although much more is left to do.

The identification of an event abstraction hierarchy is paramount for integrating not only SIGINT,⁹ but also unstructured or semi-structured OSINT,¹⁰ for instance, by fusing activity reports from both Clearnet and Darknet. By focusing on causal and aggregation relationships, this climbing of the semantic ladder is argumentative and adheres to the best practices of critical thinking [38], [39].

6. STRATEGIC THINKING

Let us now assume that a threat has been detected. The analyst now needs to estimate the intent and capabilities of attackers, and highlight issues with possible courses of action [3]. Strategic thinking is thus needed in choosing between alternative courses of action to manage APT attacks, in particular in respect to unwanted side effects that might enable the attackers to acquire information on the analyst's state of knowledge and intentions, making them aware that she is aware of their attack. For simplicity, let us consider only *deny* and *deceive*.

We therefore argue that it is necessary to explicitly acknowledge the existence of a communication link with the attackers, who will receive information about (1) our analyst's defence capabilities for detection and (2) the value of the resources she is protecting. We can thus leverage AI techniques such as Controlled Query Evaluation

⁹ SIGINT—SIGnal INTElligence—includes either individually or in combination all communications intelligence, electronic intelligence, and instrumentation signals intelligence, in whatever way transmitted.

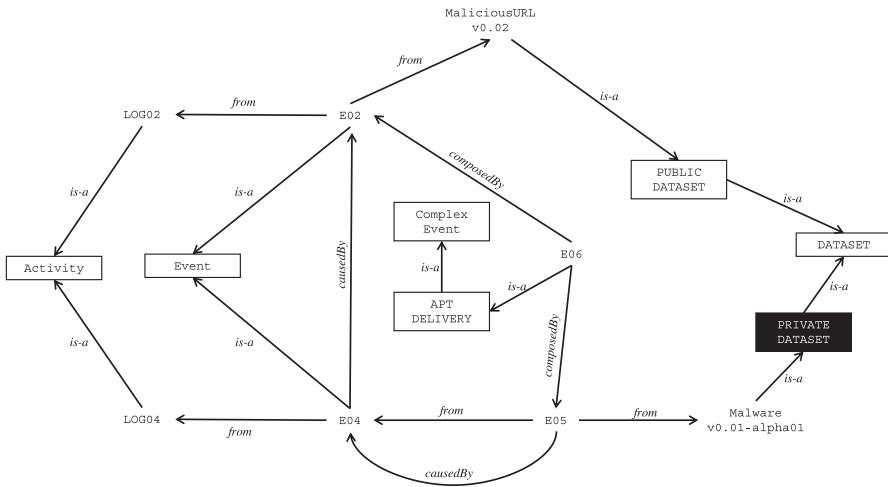
¹⁰ OSINT—Open Source INTElligence—includes media, internet (both Clearnet and Darknet), governmental data, professional publications as well as grey literature such as technical reports or preprints, commercial data, etc.

(CQE) [40], where an agent is defending some knowledge encoded in a database against an attacker who can perform queries on it. The defender can choose how to answer such queries but is compelled to obey secrecy constraints.

For instance, let us assume the analyst uses the graph database illustrated in Figure 5, where boxed labels represent classes, while unboxed ones represent objects instantiating (*is-a*) a class. We assume the existence of the *causedBy*, *composedBy*, and *from* relationships among objects; their semantics is linked to the complex event processing procedure illustrated in Section 5; for instance, E04 is *an EVENT* derived *from LOG04*, which *is an ACTIVITY*. Let us also assume that our ability to detect that the downloaded file is a trojan (event E05) is based on a machine learning algorithm trained on a dataset containing raw data about possible malware that we carefully created within our organisation, and that it is in the company's interest to keep it private. This is represented in Figure 5 by a black box (PRIVATE DATASET) since it contains *black knowledge*; that is, knowledge that should not be disclosed.

Terminating the attack by shutting the connection (*denying*) after E05 and triggering the creation of E06, which makes the analyst aware of the presence of an attack, might seem a reasonable choice. This, however, might signal the attackers that the analyst has access to superior knowledge – compared to the community – about the used malware. She needs to assume that the attackers are also able to detect E02 and E04, as the target interacted with a remote server at least partially under their control. By using, for example, CQE, over a probabilistic version of the graph database illustrated in Figure 5, we can now answer the question: what is the probability that in revealing E06 we also reveal E05? Since E06 builds on E02, which is informed by a public dataset, there is a reasonable argument suggesting that denying the attack can be explained only on the basis of publicly available information. For instance, all downloads from such URLs can be quarantined or sandboxed.

FIGURE 5: GRAPH DATABASE REPRESENTING THE DETECTION OF THE DELIVERY OF AN APT WEAPON (SEE FIGURE 4)



Inheriting uncertainty quantification about events and complex events from the previous processing and analysis steps (see Sections 4 and 5) and performing CQE over probabilistic knowledge bases makes it possible to encompass both epistemic and aleatoric uncertainties, as we showed in [41] and [42], thus providing the analyst with a computational mechanism for risk evaluation. This can also be used to derive a utility function to be used in state-of-the-art game theory approaches for choosing mitigation techniques [44], [43, Chs. 4, 5], [4]. These topics are, however, beyond the limits of this paper.

To conclude, considering at least one level into the high-order theory of mind coupled with epistemic and aleatoric uncertainty brings us closer to the real world, while at the same time revealing the complexity of the task and the need for self-aware artificial intelligence.

7. CONCLUSIONS

The threat landscape, adversarial ecosystem, and expansion of the attack surface all together link to an environment of staggering complexity where viral threats affect the entire fabric of our interconnected world. Optimising for the known threats only is not enough: we need to build resilient systems that embrace uncertainty and adapt to new types of complex attacks.

In this paper we embed defence against APT attacks into the cyber threat intelligence framework, and illustrate how self-aware AI tools can be used for building resilience and lessening the burden on the human analyst, who must always be at the centre of the design process.

We show that uncertainty-aware learning and reasoning can be an effective method for anomaly detection, which can provide human operators with alternative interpretations of data and accurate assessments of their confidence. This reduces misunderstandings and builds trust, while also reducing attackers' options for camouflage. Event processing algorithms can identify attacks spanning long intervals of time, which would remain undetected even by state-of-the-art intrusion detection systems. Finally, climbing the ladder of semantics is crucial for estimating the appropriateness of different responses to a suspected attack, and the impact (intended and unintended) of a given response.

Several avenues are ahead of us, including further experimental analyses that are already planned, but here we would like to mention one in particular that, due to space constraints, we left out, but that must be remarked upon. Although in this paper we implicitly assumed a centralised approach – that is, an analyst or a group of analysts overseeing the cyber infrastructure of a large organisation – the reality is that the staggering complexity of the task might require a more distributed approach, which can be achieved by as much as possible empowering autonomous agents at the edge of the network to collaborate with a single intent: a sort of *team of teams* [45] with the same purpose and shared knowledge. To this end, a possible strategy is to couple each analyst with an autonomous surrogate that, via reinforcement learning, could approximate the decision of its human counterpart and thus reduce even further the burden on human experts especially for the most trivial tasks.

ACKNOWLEDGEMENTS

The authors are listed in alphabetical order.

This research was partially sponsored by the Italian Ministry of University and Research via the Rita Levi-Montalcini research fellowship.

This research was partially sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of

Defence or the U.K. Government. The U.S. and U.K. Governments are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

- [1] ENISA, “ENISA Threat Landscape 2020 – Emerging Trends,” 2020.
- [2] P. Chen, L. Desmet, and C. Huygens, “A study on advanced persistent threats,” in *IFIP Int. Conf. Commun. Multimedia Secur.*, 2014, pp. 63–72.
- [3] E. M. Hutchins, M. J. Cloppert, R. M. Amin et al., “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” in *Proc. 6th Int. Conf. Inf. Warfare and Security*, 2011, pp. 113–125.
- [4] C. A. Kamhoua, L. L. Njilla, A. Kott, and S. Shetty, Eds., *Modeling and Design of Secure Internet of Things*. USA: Wiley, 2020.
- [5] H. Prunkun, *Scientific Methods of Inquiry for Intelligence Analysis*. Lanham, MD, USA: Rowman & Littlefield, 2015.
- [6] P. Pirolli and S. Card, “The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis,” in *Proc. Int. Conf. Intell. Anal.*, vol. 5, 2005, pp. 2–4.
- [7] A. Toniolo et al., “Supporting reasoning with different types of evidence in intelligence analysis,” in *Proc. AAMAS 2015*, 2015, pp. 781–789.
- [8] F. Cerutti, T. J. Norman, A. Toniolo, and S. E. Middleton, “CISpaces.org: From fact extraction to report generation,” in *Proc. COMMA 2018*, 2018, pp. 269–280.
- [9] D. Antons and F. T. Piller, “Opening the black box of ‘Not Invented Here’: Attitudes, decision biases, and behavioral consequences,” *Acad. Manag. Perspect.*, vol. 29, no. 2, pp. 193–217, 2015.
- [10] S. C. Hora, “Aleatoric and epistemic uncertainty in probability elicitation with an example from hazardous waste management,” *Reliab. Eng. Syst. Saf.*, vol. 54, no. 2, pp. 217–223, 1996.
- [11] P. S. Laplace, *A Philosophical Essay on Probabilities*. New York, NY, USA: John Wiley & Sons, 1902.
- [12] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. ICLR2015*, 2015.
- [13] D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [14] H. Wimmer and J. Perner, “Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception,” *Cognition*, vol. 13, no. 1, pp. 103–128, Jan. 1983.
- [15] C. Stoll, “Stalking the Wily Hacker,” *Commun. ACM*, vol. 31, no. 5, pp. 484–497, 1988.
- [16] S. Jajodia, P. Liu, V. Swarup, and C. Wang, Eds., *Cyber Situational Awareness: Issues and Research*. New York, NY, USA: Springer, 2010.
- [17] C. Stoll, *The Cuckoo’s Egg: Tracking a Spy through the Maze of Computer Espionage*. New York, NY, USA: Pocket Books, 1989.
- [18] J. D. Lee and K. A. See, “Trust in automation: Designing for appropriate reliance,” *Human Factors*, vol. 46, no. 1. Human Factors and Ergonomics Society, pp. 50–80, 2004.
- [19] R. Tomsett et al., “Rapid Trust Calibration through Interpretable and Uncertainty-Aware AI,” *Patterns*, vol. 1, no. 4, p. 100049, Jul. 2020.
- [20] B. M. Muir, “Trust between humans and machines, and the design of decision aids,” *Int. J. Man. Mach. Stud.*, vol. 27, no. 5–6, pp. 527–539, Nov. 1987.
- [21] V. Mavroeidis and S. Bromander, “Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence,” *Proc. EISIC 2017*, pp. 91–98, 2017.
- [22] R. Schwitter, “Controlled Natural Languages for Knowledge Representation,” in *Proc. Coling 2010: Posters*, pp. 1113–1121, 2010.
- [23] T. Kuhn, “A Survey and Classification of Controlled Natural Languages,” *Comput. Linguist.*, vol. 40, no. 1, pp. 121–170, 2014.
- [24] D. Braines, D. Mott, S. Laws, G. de Mel, and T. Pham, “Controlled English to facilitate human/machine analytical processing,” in *Next-Generation Analyst*, vol. 8758, no. 7, pp. 875–808, 2013.
- [25] D. Braines et al., “Subjective Bayesian Networks and Human-in-the-Loop Situational Understanding,” in *GKR 2017: Graph Structures for Knowledge Representation and Reasoning*, 2018, pp. 29–53.
- [26] G. Bansal, B. Nushi, E. Kamar, W. Lasecki, D. Weld, and E. Horvitz, “Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance,” in *Proc. HCOMP2019*, 2019, pp. 2–11.

- [27] J. M. Bernardo, “Bayesian statistics,” in *International Encyclopedia of Statistical Science*, Courcier, Ed., Springer, 2011, pp. 107–133.
- [28] M. Sensoy, L. Kaplan, F. Cerutti, and M. Saleki, “Uncertainty-Aware Deep Classifiers using Generative Models,” in *Proc. AAAI2020*, 2020, pp. 5620–5627.
- [29] A. Jøsang, *Subjective logic: a formalism for reasoning under uncertainty*. Switzerland: Springer, 2016.
- [30] F. Cerutti, L. M. Kaplan, T. J. Norman, N. Oren, and A. Toniolo, “Subjective logic operators in trust assessment: an empirical study,” *Inf. Syst. Front.*, vol. 17, no. 4, 2015.
- [31] S. Muggleton, “Inductive logic programming,” *New Gener. Comput.*, vol. 8, no. 4, pp. 295–318, 1991.
- [32] G. Guida and G. Mauri, “Evaluating performance and quality of knowledge-based systems: foundation and methodology,” *IEEE Trans. Knowl. Data Eng.*, vol. 5, no. 2, pp. 204–224, 1993.
- [33] X. Jin, B. Cui, J. Yang, and Z. Cheng, “An adaptive analysis framework for correlating cyber-security-related data,” in *Proc. AINA 2018*, 2018, pp. 915–919.
- [34] A. Benzekri, R. Laborde, A. Oglaza, D. Rammal, and F. Barrère, “Dynamic security management driven by situations: An exploratory analysis of logs for the identification of security situations,” in *Proc. CSNet 2019*, 2019, pp. 66–72.
- [35] M. R. Vilamala *et al.*, “A hybrid neuro-symbolic approach for complex event processing (extended abstract),” 2020.
- [36] T. Xing *et al.*, “Neuroplex: learning to detect complex events in sensor networks through knowledge injection,” in *Proc. SenSys2020*, 2020, pp. 489–502.
- [37] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, and L. De Raedt, “DeepProbLog: Neural Probabilistic Logic Programming,” in *Proc. NIPS2018*, 2018, pp. 3749–3759.
- [38] N. Hendrickson, “Critical thinking in intelligence analysis,” *Int. J. Intell. CounterIntell.*, vol. 21, no. 4, pp. 679–693, 2008.
- [39] D. Walton, *Fundamentals of Critical Argumentation*. Cambridge, UK: Cambridge University Press, 2005.
- [40] G. L. Sicherman, W. De Jonge, and R. P. Van de Riet, “Answering queries without revealing secrets,” *ACM Trans. Database Syst.*, vol. 8, no. 1, pp. 41–59, 1983.
- [41] F. Cerutti *et al.*, “Obfuscation of semantic data: Restricting the spread of sensitive information,” in *Proc. DL2014*, 2014, pp. 434–446.
- [42] F. Cerutti *et al.*, “When is Lying the Right Choice?,” in *Proc. 1st Workshop on Lang. and Ontologies*, London, UK: Association for Computational Linguistics, 2015.
- [43] E. Al-Shaer, J. Wei, K. W. Hamlen, and C. Wang, Eds., *Autonomous Cyber Deception*. Switzerland: Springer International Publishing, 2019.
- [44] J. Pawlick, E. Colbert, and Q. Zhu, “A Game-Theoretic Taxonomy and Survey of Defensive Deception for Cybersecurity and Privacy,” *ACM Comput. Surv.*, vol. 52, no. 4, Aug. 2019.
- [45] S. A. McChrystal, T. Collins, D. Silverman, and C. Fussell, *Team of Teams: New Rules of Engagement for a Complex World*. New York, USA: Portfolio/Penguin, 2015.
- [46] C. Pace *et al.*, *The Threat Intelligence Handbook: A Practical Guide for Security Teams to Unlocking the Power of Intelligence*. CyberEdge, 2018.
- [47] R. L. Ackoff, “From data to wisdom,” *J. Appl. Syst. Anal.*, vol. 16, no. 1, pp. 3–9, 1989.