ACIG APPLIED CYBERSECURITY & INTERNET GOVERNANCE

NASK

Artificial Intelligence for Cybersecurity: Offensive Tactics, Mitigation Techniques and Future Directions

Received: 28.09.2022
Accepted: 02.11.2022
Published: 04.11.2022

Erwin Adi Deloitte Risk Advisory Pty Ltd, Australia, ORCID: 0000-0001-7120-1967
Zubair Baig Deakin University, Victoria, Australia, ORCID: 0000-0003-0557-1550
Sherali Zeadally College of Communication and Information, University of Kentucky, USA, ORCID: 0000-0002-5982-8190

Cite this article as:
E. Adi, Z. Baig, S. Zeadally, "Artificial Intelligence for Cybersecurity: Offensive Tactics, Mitigation Techniques and Future Directions", ACIG, vol. 1, no. 1, 2022, DOI: 10.5604/01.3001.0016.0800

Corresponding author:
Erwin Adi; Deloitte Risk Advisory Pty Ltd, Australia;
ORCID: 0000-0001-7120-1967;
E-mail: eadi@deloitte.com.au

Abstract

Cybersecurity has benefitted from Artificial Intelligence (AI) technologies for attack detection. However, recent advances in AI techniques, in tandem with their misuse, have outpaced parallel advancements in cyberattack classification methods that have been achieved through academic and industry-led efforts. We describe the shift in the evolution of AI techniques, and we show how recent AI approaches are effective in helping an adversary attain his/her objectives appertaining to cyberattacks. We also discuss how the current architecture of computer communications enables the development of AI-based adversarial threats against heterogeneous computing platforms and infrastructures.

Keywords

adversarial AI, cyber infrastructures, data analysis, supply chain compromise

1. Introduction

Artificial Intelligence has enabled cybersecurity researchers and practitioners alike to design and develop cutting-edge solutions to counter the ever-expanding and increasingly sophisticated types of cyberattack that threaten contemporary computing systems and platforms. Increasing production and marketing of AI-based cybersecurity solutions have set the trend during the past decade [1, 2]. Recent advances in the AI research domain have empowered cybersecurity systems to manage machines autonomously, and to safeguard these by creating rapid defence and reprisals against an adversary, in near real-time [3]. On the other hand, the adversary has also gained significant potency and far outreach in his/her attack strength owing to the same advancements

in Artificial Intelligence technology. Though the core attacker steps comprising a data breach, namely vulnerability detection, exploitation, post-exploitation and data theft [1], remain the same, the potential impact of an AI-based system deployed to do so is of increasing concern to all. This is due to the shift from traditional (Fig. 1.) to modern Internet architecture (Fig. 2.).

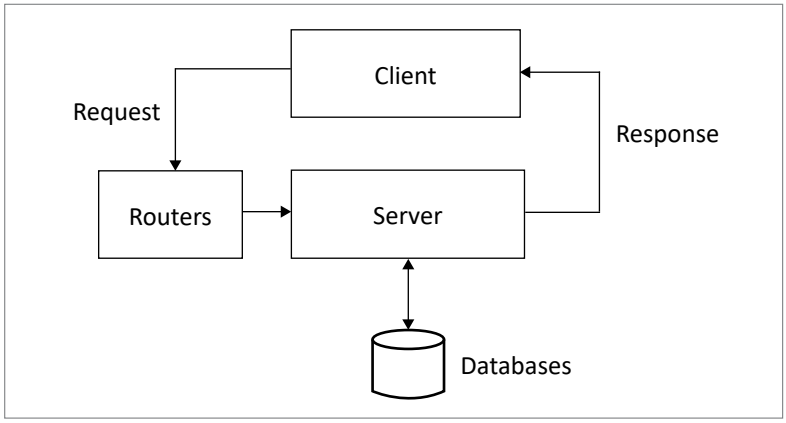

Figure 1. The architecture of the Internet.

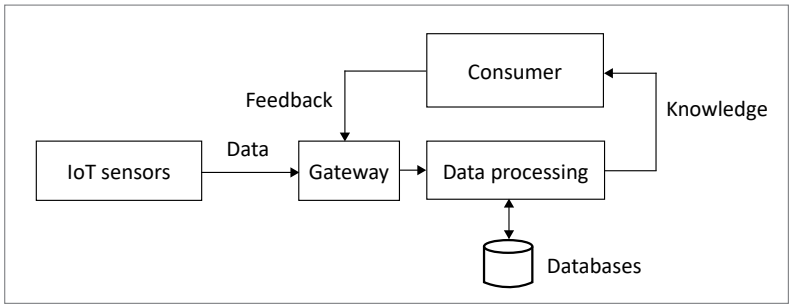

Figure 2. The architecture of the modern Internet, adapted from [4].

The Internet was traditionally viewed as networked interconnections of client-server computers as Fig. 1. shows. The client, such as a PC, sends a request packet to a server. The server processes the request, initiates some actions such as fetching data from the database, and sends its response back to the client. These computers are interconnected by network devices (e.g. routers). Such a client-server model facilitates the exchange of data, but not knowledge or insightful information that has been processed by an AI machine.

However, the modern Internet needs to be modelled as a communication system not only for exchanging data, but also to render feedback, and knowledge [4] as Fig. 2. shows. A large amount of data is generated by IoT devices, through their sensors that generate phenomena data including user or device location, speech patterns, text, emotions (e.g. through a like button), social links, pictures, and videos. These IoT devices send the data to other devices, including to the Cloud. Their interconnection is served by a gateway that supports heterogeneity in transmission techniques and communication standards. At the other end of the communication line, a machine is responsible for the processing of collected data to improve its usability. It generates outputs from the classification of the data to location recommendations (e.g. a *Google* map route recommendation). The machine either stores the data in the database or transmits the same (or data converted into knowledge) to a consumer, such as to a smart device, or a monitoring system. This chain of devices collectively comprises an intelligent system, which allows for iterative feeding of data to learn adaptively from sensor data and through device feedbacks.

As a computing device of an intelligent system, each of the resources illustrated in Fig. 2. (i.e. IoT sensors, gateway, data processing, and consumer) can become a target of AI-based cyberattacks (Fig. 3.). They are vulnerable to two attack artifices: those crafted by rational agents or bots, and those that comprise behaviour-mimicry attacks. These two artifices cover the definition of AI, i.e., agents that act in a human or rational way [5]. The first artifice, acting in a human way, invokes the Turing Test wherein human observers cannot distinguish whether the behaviour of a system was caused by either a human or a bot. The second artifice, of acting rationally, means that a rational bot can yield an optimum solution given a complex challenge and offering a wide range of corresponding solutions with varying degrees of risk.

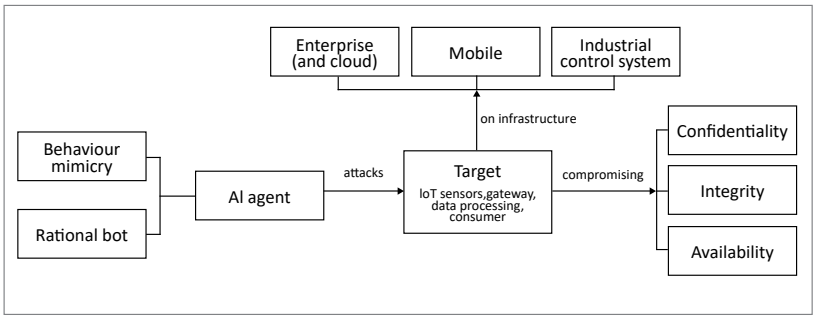

Figure 3. Techniques attacking the modern Internet.

A target operates in one of the three computing domain infrastructures, namely, enterprise (including Cloud), mobile, and industrial control systems [6]. Fig. 3. shows that targets become victims when either one of the security goals is compromised. These are the confidentiality, integrity and availability of a computing system. Thus, an intelligent system becomes a victim when any one of the targets (i.e. IoT sensors, gateway, data processing, consumer) that is part of a computing infrastructure (i.e. enterprise/Cloud, mobile, industrial control systems) is attacked by some AI artifices (i.e. behavioural mimicry, rational bot), with the effect that one or more of the security goals (i.e. confidentiality, integrity, availability) is compromised. This means that attackers can employ a rational bot to advise on an optimum tactic flow out of many attack possibilities that have been described above. When engaging in a specific technique, the intelligent agents can deliberately find intrusion actions that produce data so as to get misclassified as normal. Thus, malicious AI agents capable of discovering the weakest link in a cyber system can be designed to launch adversarial AI attacks.

The work we present in this paper offers in-depth analysis of adversary approaches which exploit AI techniques to launch sophisticated cyberattacks. The approach allows us to see how the modern Internet phenomenon (Fig. 2.) and the complexity of the cyber kill chain [7] in penetrating cyber infrastructures can lead to the emergence of AI agents attacking the latter (Fig. 3.). We show that:

- as the modern Internet exchanges feedback data and knowledge, in addition to IoT-generated data, current devices are vulnerable to supply-chain compromise;
- the combination of possible tactics seeking to infiltrate cyber infrastructures is too complex to allow human analysts to understand zero-day attacks manually;
- in adversarial AI, agents can leverage the abundant data and the complexity of the problem domain.

1.1. Comparison with previous works

Discussions [8–10] to be found in the literature presented the Internet as a client-server architecture (Fig. 1.). Attacks were categorised on the basis of the control and modification of request packets [8]. From this point of view, adversarial AI techniques were tested only in respect of the meeting of data analytics goals, including as regards

the efficacy of an intrusion-detection system and the ways in which network traffic can be misclassified [11, 12], or in relation to how a DoS attack is carried out through control of the volume of request payloads and their corresponding packet sizes [12, 13]. Adversarial AI was viewed in terms of its being a matter of finding data models to detect phishing or credit-card fraud [12, 14], rather than having an external IoT device manipulating the model. Malware was analysed using white-box approaches [11, 14, 15], rather than being seen from the point of view of a rational bot that can combine previously-known techniques from a knowledge base. Analyses of AI attacks on intrusion-detection systems were viewed from a one-sided perspective [16] wherein datasets can be manipulated to produce attack data. This view alludes to the IoT perspective (as Fig. 2. shows) whereby response packets also act as knowledge to create adversarial data.

Other works [17–20] also fail to present background on how AI agents can leverage supply-chain compromise to target intelligent data-processing machines deployed on a range of infrastructures including autonomous systems and critical infrastructures. They have shown how adversarial AI agents can be used mostly for data-processing systems or for specified cyber infrastructures, as Tab. I. shows.

Table 1. Comparison with previous works.

Supply Chain	Supply Chain	Data Analysis
Other papers	[21]	[11–20]
Our contribution	*Covered*	*Covered*

Although the literature has discussed the circumstance that AI can be used for both attack and defence, few have examined the use in adversarial cyberattacks [21]. Adversarial AI techniques have allowed for the development of video games and natural language understanding [22], speech recognition, computer vision, online recommendation systems, and bioinformatics [23]. Most techniques that discussed cyberattacks observed from the data-analysis point of view (Tab. I.), within which AI models were challenged by reference to the ways in which request packets can be manipulated, rather than how external devices can infiltrate AI behaviours. In that circumstance, there was little discussion of ways in which AI has gained use in current adversarial cyberattacks, or indeed on methods developed to mitigate intelligent agents designed for such attacks.

2. Adversarial AI techniques used in Cyberattacks

Artificial Intelligence (AI) techniques range from mathematics and statistics to logic models, whereby procedures are encapsulated in an algorithm. The latter are known commonly as machine-learning algorithms, comprising both machine learning and AI interchangeably. Machine-learning algorithms analyse data in terms of samples and features. As an illustration, if a dataset is in the form of a table, the samples are the rows, and the features or dimensions are the columns.

We briefly introduce certain common AI techniques in the following paragraphs, before going on to discuss how they might be put to adversarial purposes.

Expert Systems represent one of the earliest computing techniques for decision-making. By way of a series of if-then-else flows, human experts are mimicked in reaching a final state, given a range of input data. In cybersecurity, such can serve as a knowledge base identifying asset vulnerabilities [24].

In turn, Particle Swarm Optimisation approaches [25] mimic the behaviour of social animals, in that each individual learns effectively from the others, with a view to optimum solutions being arrived at, e.g. as regards food. Such techniques were used for classification, weight optimisation, feature selection and dimensionality reduction [26].

Naïve Bayes [27] is a classic algorithm that gives acceptable results as data are classified. As such, it is used as a benchmark in comparing classification performance when a new machine-learning algorithm is developed.

Support Vector Machines [28] is a classification technique that can classify non-linear data. It is relevant to cybersecurity analysis because Internet traffic consists of heterogeneous data generated from a wide range of devices.

Artificial Neural Networks (ANNs) [29] analyse all features from each sample as a meshed network. This allows for flexible re-learning of new samples even after the model converges at the end of the training phase. This behaviour of ANNs is suitable for the analysis of Internet traffic in which there are rapid changes of the data pattern.

While Naïve Bayes and SVMs process the values directly from data features, ANNs can have multiple layers of intermediate features. Each layer can be designed to represent a set of features that is derived intuitively from the previous layer. Such layered networks are used in Deep Learning.

Deep Learning or Deep Neural Networks (DNNs) come with their derivatives, each with variations as to how networks are connected [30]. Their applications are discussed further in Section 2.1. Deep Learning networks have at least an intermediate, or a hidden, layer of units that is present between the input and output values. When the flow which adjusts the weight progresses in one direction from the input to the output layer, the network is called a feedforward network. If the adjacency of input values matters, then Convolutional Neural Networks (CNNs) become the architecture of choice. Instead of having a mesh of connections from the input values to the following layer as in DNNs, each unit in the CNN hidden layer is connected to a group of input values. As such, adjacent input values are captured as a spatial region. While CNN architectures represent spatial relationships, Recurrent Neural Networks (RNNs) consist of an architecture by which to model data with temporal characteristics. RNN outputs at time τ are looped back, such that when unfolded its value can be calculated together with the input at time $\tau + 1$. On this basis, the network remembers and computes by reference to inputs from a series of time states. A derivative of RNN architecture is provided by Long Short-Term Memory (LSTM) networks, which can handle a longer chain of units without losing prior information. A Generative Adversarial Network (GAN) is a pair of ANNs wherein one network generates fake data samples that mimic the original training data, and the other network classifies the fake and original data. The two networks compete during the training iterations, with one network attempting to mislead the other. Thus, the generator aims to create fake samples that the discriminator cannot distinguish from the training data. The discriminator aims to classify the fake samples from this training data. As this section will discuss, GANs have attracted much attention in adversarial AI techniques.

2.1. Adversarial AI agents

Cyberattacks can leverage defensive AI techniques to compromise cyber systems. There are two characteristics of modern AI solutions that allow for the emergence of malicious AI agents, i.e. iterative learning and the use of a knowledge-base. Iterative learning allows devices to learn from the data generated from other ata-processing devices. As an example, defensive techniques such as anti-malware can be repurposed to develop new variants of mobile malware through iterative learning. In [31], the authors used Deep Learning to ascertain whether a malware variant was detected by anti-malware. The neural network iteratively mutated the variants by obfuscating their code until it was able to evade a group of anti-malware programs tested. Similarly, in [32], the authors used Genetic Programming to mutate executable programs. In this case, the subroutines of the programs constituted the chromosomes. They were selected and crossed over to create new malicious code, and then the resulting code was

obfuscated. To test whether the code had become malicious, anti-malware was used twice, i.e., before and after the code was obfuscated. Iterating this process improved the selection of fitness values, in that a smaller number of malware detections was noted after the second test was conducted, as compared with the first one.

Inevitably, the use of a knowledgebase can also give malicious AI agents a competitive advantage. The authors of [33] described two types of real-life AI-based attacks that had occurred previously, i.e. attacks that took advantage of humans as the weakest link; and those that benefited from a rich knowledgebase as Fig. 4. illustrates. Using humans as the weakest link, in [33], attackers employed a tool to observe how a human user clicked and forwarded messages on social media. This allowed attackers to identify the most vulnerable target prone to a clicking-based phishing attack, and employed AI-based techniques to tailor highly relevant messages to the targets. In the second type, where the attack employed a knowledgebase, it was possible to describe a software vulnerability that had previously been proven to patch software. In a competition setting, the knowledgebase was employed to create autonomous attacks targeting and successfully compromising software systems belonging to other contestants. In [33], the authors also described a hypothetical case in which AI agents launch cyberattacks (Fig. 4.). It shows that worms, or codes that can spread autonomously to other systems, can automate the above cyberattack scenarios, e.g. by creating phishing emails or crippling target systems.

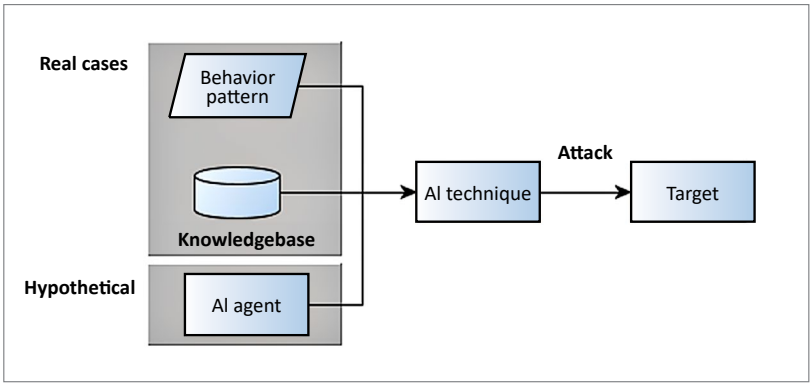

Figure 4. The current view on how AI techniques can be employed to launch cyberattacks.

Our study agrees with the above view that AI-assisted cyberattacks are increasingly a threat as they can conveniently circumvent existing security controls. Intelligent agents can engage in the autonomous targeting of weakest links in system, mimic legitimate behaviours, bypass intrusion-detection systems, and spread across different infrastructures. In this section, we show that current research has developed certain scenarios previously regarded as hypothetical. Fig. 5. illustrates the structure of the remaining discussion.

Fig. 5. shows that malicious AI agents launch cyberattacks through behaviour mimicry and rational bot techniques. As a result, the attacked target behaved differently in terms of computing output or performance. The AI agent captures these differences to optimise its attack strategy. The attack strategies are further applied to evasion, data poisoning, and model stealing techniques.

2.2. Evading detections by mimicking legitimate behaviours

Cyberattack detection has been described as detecting anomalous behaviour in networks or by users [34, 35]. Intelligent agents would mimic normal behaviour of networks, computer systems, or users, in order to bypass intrusion-detection systems. These agents are equipped with the statistical distribution of human-generated traffic patterns when online [34]. Intelligent machines would mimic the action

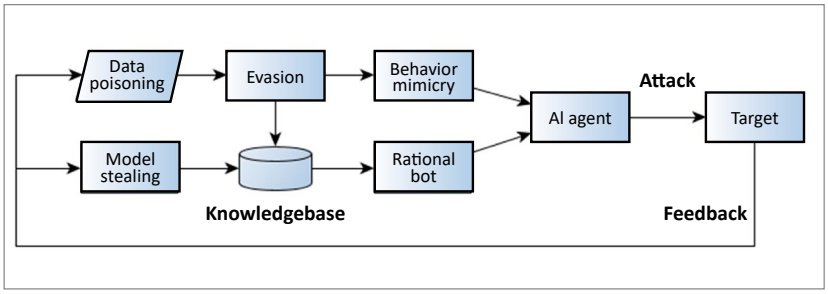

Figure 5. Intelligent agents launch cyberattacks through behaviour mimicry and rational bot techniques.

of a human very closely [35]. Hence, the capability of an adversarial intelligent agent to evade detection can be attributed to behavioural mimicry.

The mimicking of legitimate behaviours can be made possible because the data generated by a device are no longer a mere response to a request packet as Fig. 1. shows. In today's Internet architecture (Fig. 2.), intelligent devices process request packets to generate insightful information (knowledge) in line with a device's data-processing behaviour. Thus, for example, a cloud service might send knowledge to a smartphone about which communication path would be optimum to traverse between two points.

The knowledge depends on the data-processing intelligence in observing the smartphone user's behaviour, e.g. as regards time of day, mode of transport, and most apt user preference between shortest path and journey time.

The data generated by a device thus depend on what that device has learned in addition to the request packet. It is therefore possible to train a device by mimicking some normal behaviour patterns in order to generate certain targeted data. This means it is possible to feed false data to the Cloud service (Fig. 2.) as generated from IoT sensors (Fig. 2.). Such a technique is known as the "supply-chain compromise" [36–38]. Behaviour mimicry is therefore an approach that can be taken in launching supply-chain compromises.

2.2.1. Data poisoning

The techniques to evade detection systems poison a system's input data. Poisoned data is basically contaminated data that can cause the detection system to misclassify inputs. Data poisoning assumes that the adversarial system has *a priori* knowledge of normal patterns. For example, in Internet traffic, normal network traffic patterns are those generated by human users as they browse websites. In [34], the authors demonstrated attacks that pre-empt a target's service successfully evading an intrusion-detection system even as the target was flooded with normal traffic, with the target caused to drop packets. Fig. 6. offers a statistical illustration of poisoned data. The black curve represents the distribution of a normal traffic feature value; the grey bar/curve represents attack values; and (left figure) the red threshold shows a detection system, which separates attack data from normal data. The right figure illustrates data poisoning. The normal values are contaminated with the attack values, evading the threshold bar, and rendering the attacks stealthy.

2.2.2. Stealthy attacks

Fig. 6 summarises the method by which stealthy AI-based cyberattacks can be designed, i.e. through the supply of data whose anomalous value range overlaps with the acceptable range. While the above example [34] demonstrated that stealthy attack traffic can be prevented by the dropping of packets, the authors of [40] showed how dropped packets in wireless networks can be made stealthy, causing the invariable blacklisting of legitimate nodes. In wireless networks, nodes (e.g. wireless

devices) communicate within a certain communication range defined through by way of respective radio power values. Intrusion-detection systems can be deployed by means of collaborating nodes forming a series of overlapping radio range for scanning, which allows a node to cascade its network range of observation alongside neighbouring nodes. A malicious node can therefore be detected when it fails to forward packets within a time threshold to its neighbour. However, invoking set theory, the authors of [40] demonstrated that a malicious node located at an intersection of two sets of radio ranges can intentionally misroute packets such that they are forwarded to a victim node, causing the latter to drop packets and be blacklisted by the intrusion-detection system. This shows the way in which a malicious node that had learnt about the threshold value of a system and its position was able to affect the reputation of another node.

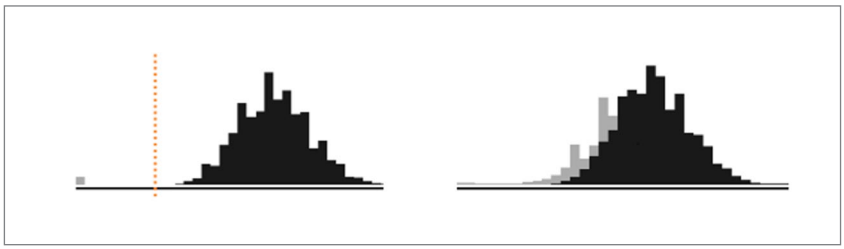

Figure 6. Left: a threshold (dotted vertical line) separates attack values (grey) from normal values (black). Right: attack values mimic the distribution of normal values. Adapted from [39].

Stealthy attack methods are applicable to a wide range of cyber infrastructures, such as industrial control systems, facial recognition, and autonomous vehicles. In industrial control systems that monitor the degree of acidity (pH) of water, attacks can change the water pH values to a dangerous level [41], where the adversary is assumed to have compromised the pH meter device. The authors of [41] showed that a detection system that depends on a threshold value can be evaded by having the attacker adapt to the threshold value/range.

A case of the use of supply-chain attacks can be seen in the protection of cyber-infrastructures in which use is made of certain physical barriers, with a view to physical intruders being delayed, deterred and detected. Physical access, e.g. involving direct access to cyber-equipment, contributed to 56.3% of attack vectors in 2019 [42]. The mitigation of physical intrusions would entail installation of physical locks to deter and delay access, or cameras to detect presence. In this case, facial recognition can play a role in protecting cyber-infrastructures. Cameras can be programmed to recognise faces and raise alarms when they capture non-whitelisted facial images. However, in this case it is possible for cameras to be evaded to raise an alarm. Evading a classifier that recognises images of a person's face is to be done where an indiscernible image (such as a eye glass) is added to the input image [43], causing the classifier to recognise it as a different person. The authors of [43] first assumed the possession of the knowledge of the classifier, so that stealthy patterns might be designed. They used Deep Neural Networks to construct the classifier and to find a set of patterns, r, of the left-tail norm x, such that $x + r$ is classified into a desired class. Second, the study conducted a black-box test by applying the pattern (i.e. the eye glass) to a commercial face-recognition Cloud-based solution. Thus, the test relied on the software output as feedback to readjust the stealthy pattern r. However, the cloud classifier only outputs the top three classes, causing further difficulty with finding an indiscernibly modified input $x + r$. The study therefore used the Particle Swarm Optimisation algorithm that presented intermediate personified images, allowing each iteration to move away from the previous solution space and to approach the candidate solution more closely. As a result, 19 out of 20 images tested in the study proved to be evaded successfully.

Knowing the a threshold value of how much friction and wind can be tolerated by autonomous vehicles allows adversaries to fake a vehicle's positions without being detected. The authors of [44] demonstrated that adversaries can infer the total number of errors that can be tolerated by two target autonomous vehicles. The targets employed the Kalman filter, applied commonly in estimating the position of remote vehicles from the time lapse following on from the last collection of data. If an attacker compromises the control input/output of an autonomous vehicle, then the time-series data describing the vehicle's position can be derived, such that the error window size can be inferred. This allows the attacker to possess knowledge as to how much total error was tolerated within a time window. Injecting such stealthy errors into the controller can cause vehicles to deviate from their positions. As a result, a drone was, for example, shown to take a 50% longer time to accomplish a mission, while a rover took 30% longer [44]. In addition, the authors demonstrated that a large, poisoned dataset successfully evaded an intrusion detection system when it was subjected to drone memory, causing a drone to deviate 11 degrees when landing – a departure sufficient to prove fatal.

2.2.3. Perturbation

Another name for data poisoning is perturbation attack, as found in common use against industrial control-systems infrastructure. Perturbations are input noises whose range values are permitted by the system. In a network of water pipelines, the intentional perturbing of a water-meter reading can cause a machine-learning-based classifier to allow unusable water to be distributed to the population. A study [45] proposed that such a classifier should behave as a linear constraint to detect anomalies, given its mimicking of the water-flow behaviour in a pipeline network. While the water network in question was equipped with meter sensors by the provider, a linear constraint mandates that, for example, linear values are expressed as: *Meter1* > *Meter2* + *Meter3* is a legitimate pattern. Otherwise, the classifier detects an anomaly. The water network that the study observed was complex, consisting of 51 attributes, i.e., the values from 25 meters and the state of 26 actuators (i.e. valves that control flow), yielding a complex linear matrix. The study leveraged the system's tolerance for noise and normal fluctuations to perturb water-flow measurement, and ensure successful evasion where the detection of bad water flow was concerned. Similarly, perturbation of the voltage in an electricity network system can cause the system's classifier to misclassify electricity events identified in network traffic flowing into the grid [46]. In both cases, i.e. a network involving water [45] and electricity [46], the adversary can take an extra step in generating a normal pattern by assuming the classifier model. As is noted above, in the case of a water network, an adversary can assume a linear-constraint classifier, whereas in an electricity network, the adversary can assume a Convolutional Neural Network classifier, given that electrical-event behaviours are described ideally in time and space. Furthermore, the studies of both the water and the electricity network assumed the compromising of the systems' meters by an adversary, with this allowing it to measure results obtained through system perturbation, and to generate poisoned data.

A variation for an adversary in modelling the target system is to assume feature ranking. In [47], the adversary perturbed the feature values describing events in an electricity network system, starting from the highest-ranked features. The perturbation of feature values could be achieved iteratively to the next ranked feature, until such time as misclassification of electricity events was observed, with this therefore aiding the adversarial objective of disruption/sabotage.

2.2.4. Fuzzing

The poisoning of input data can rely upon fuzzing techniques applied generally in software auditing. Fuzzing generates many input patterns that are input to a software program, so that many execution paths can be monitored for the purpose of bug-detection. The black-box fuzzing approach is dynamic analysis, by which poisoned inputs are fed into a program and the behaviour thereof is monitored,

Table 2. AI techniques are used at all fuzzing stages.

Fuzzing stage	Challenge	Non-AI solutions [49]	AI solutions [50]
Seed file generation	Find the pattern that can save CPU time	Standard benchmarks; open-source samples	LSTM to learn known vulnerabilities in the sample
Testcase generation	Cover more program execution states	Dynamic taint analysis; probabilistic context-sensitive grammar	Neural networks to predict parts of greater vulnerability
Testcase filtering	Select inputs likely to find new paths	Hardware (Intel)	CNN to predict reachability of inputs
Operator selection	How to change input patterns efficiently	Generation-based: knowledge of program input is required	Mutation-based, derived from genetic algorithm
Exploitability analysis	Find vulnerabilities, not only crashes	Random-based fuzzing strategy	SVM and Bayesian to analyse features

to confirm whether one of the security triads (i.e. confidentiality, integrity, and availability of the program) has been compromised. In [48], the authors describe an example in which a black-box fuzzing approach has been used to ensure that a program cannot be executed. Poisoned files are used as input to evade a program's parser recognising malicious files, with the result being for the program to execute the file. Test files were poisoned by fuzzing their bytes (i.e. random byte, delete, clone, or overwrite) from certain sections (e.g. the header). The poisoned files were still accepted for execution by the program, with the result that it was crashed successfully.

AI techniques have played a significant role in exacerbating the difficulties noted previously in fuzzing steps. As Tab. II. shows, survey studies [49, 50] make it clear how each fuzzing step poses its own challenges and solutions. Fuzzing requires knowledge of how a target program is coded, and how it behaves under certain test patterns. As there are many cases to test, good initial patterns, or seed values, are required to efficiently find new execution paths to save computational resources (e.g. CPU time). Fuzzing is therefore an optimisation problem with multidimensional input vectors. Traditionally, certain assumptions have been made as to which input vectors can reveal program vulnerabilities efficiently; and some open-source seed patterns were adopted in consequence. The detection of new patterns capable of revealing vulnerability in software execution paths was achieved through random change of input vector values. In [50], the authors presented a survey reviewing 44 studies showing how AI techniques have the advantage of processing data as vectors, allowing many input patterns to be trained and labelled. Seed files can be represented as feature vectors, and good input patterns can be learned through training. Through the adoption of certain mutation-based algorithms, new feature vectors can be generated efficiently. These features gain analysis as AI, SVM, Bayesian and other strategies are used to select the fittest input values.

2.2.5. Discussion

We can make three observations from our analysis of the behaviour-mimicry techniques present in various IT infrastructures, namely that:

- normal patterns can be learned, whereas where behaviour mimicry is absent, they are either assumed, simulated or captured to create a dataset;
- data poisoning represents a subset of evasion attacks;
- the said mimicry of behaviour expands to mimicking machines.

Expanding on these points we first note how the adversary is in a position to learn. The distribution of normal patterns does not have to be assumed. Fig. 5. illustrates this, with the feedback arrow making this clear. In the modern

Internet, target systems can be intelligent devices (e.g. smartphones) that send feedback information on receipt of a request message. Such feedback data are useful for the adversary to learn the threshold value of a detection system, and to eventually create a normal pattern dataset. For example, in autonomous vehicles, an attacker can infer the errors that the vehicle can tolerate by learning from its last location [44]. In industrial control systems, attackers can infer the tolerated noise in a water [45], or by observing water/electricity flows measured by the compromised meters in an electrical system [46]. Thus, data poisoning techniques can learn and create data, rather than merely assuming normal patterns.

Second, the goal of data poisoning in cybersecurity is the evasion of detection. This differs slightly from how attacks were defined outside the cybersecurity context. From a data-analysis perspective, poisoning attacks take place at the training stage, while evasion attacks occur at the testing stage [17, 48]. This reflects the differing data-analysis strategies, i.e. white-box and black-box. As attackers in a white-box setting have knowledge of the AI model training stages, training data can be poisoned, and inputs classifiable as false negatives can then be supplied at the testing stage. On the other hand, cyberattacks should be viewed from the black-box perspective. As the authors of [51] discussed, attackers in a black-box setting can estimate the allowed data values for poisoning, by learning from the feedback data. This agrees with the first observation mentioned above, that normal patterns can be learned/estimated.

Third, AI-assisted cyberattacks are concerned with mimicking, not only human behaviour, but also machine behaviour. Techniques applied in evading intrusion-detections mimic, not only human browsing behaviour [34], but also machine behaviour involved in enterprise network systems [40], industrial control systems [41], and autonomous systems [44]. The definition of intelligent systems may therefore need expanding, to include systems that can convince other machines of their status as legitimate peers.

2.3. Rational bots

In the modern Internet architecture, AI-assisted black-hat agents enjoy an incomparably greater advantage over white-hat human analysts. Today's Internet creates a network of networks too complex for manual traffic data analysis to be performed. Traffic analysis has become even more complex because devices are interconnected with others across various heterogeneous infrastructures. Illustrating this, in [52], the authors proposed a system that detects whether an elderly person has fallen accidentally. This system consists of a wearable device (mobile) connected to a digital gateway at home with a view to (remote) healthcare of the elderly being facilitated. The gateway sends data to a Cloud service that determines whether the device user has fallen accidentally at home, or merely taken a rest-motivated lie-down in a deliberate and intentional way. When a true fall is detected, the Cloud service sends an alert message to family members and to an emergency centre (critical infrastructure). This illustration shows that complex and diverse IT infrastructures are involved in providing a timely, critical solution. White-hat human analysts can become overwhelmed when attempting to find patterns from intrusion logs across different infrastructures and platforms. As such, they have built knowledge bases to model cyberattacks [53]. Such bases allow cybersecurity analysts to share their knowledge about how new techniques and tactics have succeeded in compromising a target, and allow them to engage in incident handling. However, such knowledge bases can also be leveraged by black-hat AI agents, who would like to build rational bots capable of discovering security holes in a complex system whilst focusing on the weakest link. Such a tactic for discovering security holes that are outside the current compromised system is known as "lateral movement" [54, 55, 56]. AI agents can be used by adversaries to engage in rational detection of a system's weakest link, with lateral-movement attacks then performed.

2.3.1. Model stealing

Model stealing is a technique by which to create a model mimicking algorithms adopted for detecting attacks, as Fig. 7. shows. A defensive classifier for detecting attacks can classify its input data to attack/legitimate as output. Both the input and output data-vectors create a new dataset for an attacker. The output data acts as the label to each input, such that the new dataset is a labeled one capable of being used to train AI techniques. The attacker can use this new dataset to train an adversarial model mimicking a legitimate intrusion-detection system. Deep Neural Network techniques are adopted as an adversarial model to mimic a machine learning-based classifier. In [57], the authors used a classifier for machine learning-based intrusion-detection, and the proposed stealing model yielded results of 99.59% accuracy.

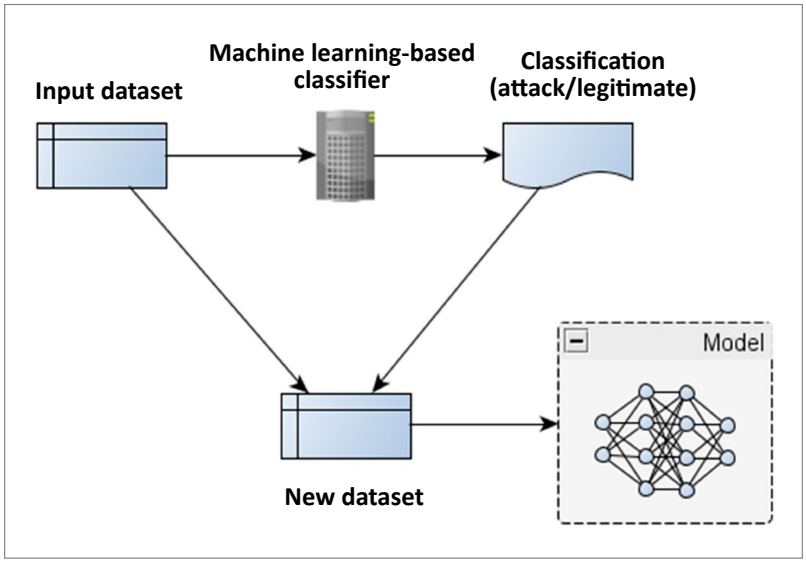

Figure 7. Model stealing, adapted from [45].

As Tab. III. makes clear, most challenges in model stealing appertain to knowledge of input data accepted by the victim service e.g. a machine learning-based classifier, so that the adversary knows what to query. On one hand, exact input data is required for the classifier to be stolen; but on the other there is a need for knowledge on classifier, in order for acceptable input data to be produced. Construction of that input data entails estimation of its range of parameters in a multidimensional space, as well as limitation on the amount of required data samples with a view to computational complexity being minimised. Input-data generation is one of the issues faced by AI systems. To address this, the author of [58] adopted a smaller external dataset to estimate the input range of a dataset used by a victim service hosted in the Cloud. Generative Adversarial Networks (GANs) were deployed to estimate parameters in the stolen model, with the external dataset as the input. The adversary had no *a priori* knowledge of how much of the external data overlapped with the input data in the Cloud. To overcome this, a knowledge distillation technique [59] was adopted to measure the loss, i.e. the difference between the output from the GAN model and the output from the Cloud model. Smaller differences denote correct estimation of output values. Then, the output data that the victim was able to label correctly label received a higher weighting than other output data. This allowed GANs to estimate the parameters in the stolen model, thereby generating synthetic input data. In return, the latter yielded a better stolen model. The quality of the input data was measured by reference to the inception score-higher numbers come from a lower entropy value when the joint probability of the synthetic and victim data is high. The work showed that synthetic data achieved 60.58% higher inception-score values than the external data.

Table 3. Model stealing issues depend on where the data is processed.

Location of data transformation activity	Challenge	Solution
Edge	Synthesising input Data	Minimising [58] or maximising [60] the entropy of joint probability distribution of victim-synthetic
Cloud	Minimising the amount of traffic for query	Finding a subset of input data [62]
IoT (mobile)	Noisy media, incomplete classification data	Reconstructing the data employing human judgment in viewing images [63]

Similarly, the authors of [60] employed an external dataset and the knowledge-distillation technique to create synthetic data. In contrast to the work of [58], this seeks to maximise the difference between the victim's output data values and the output values obtained from the stolen model to measure loss. The measure employed zeroth-order gradient estimation [61] to update the weight that maximises the entropy of probability distribution between the victim and stolen model output. By training both models an input dataset was generated which was then used in training the stolen model. The synthetic dataset maximises the learning of the stolen model, and therefore creates a highly accurate stolen model.

When the machine learning-based classifier (Fig. 7) is located in the Cloud, the challenge of stealing the model is to query the Cloud service discretely. Many queries to the cloud can trigger an alarm. In [63], a discrete querying of the model was developed by employing the transfer-learning technique, which finds a subset of the input data such that the total number of samples is minimised. With DNN, input data was pre-trained to yield a new dataset of lower dimension. The authors of [62] demonstrated that the stolen model of a Cloud-based image classification model achieved accuracies of 83.73%. The number of queries was 1290, compared with 5000 for other comparable results, which suggests that the model was stolen discreetly.

When the machine learning-based classifier is located on an IoT platform, the challenge of model stealing involves handling of incomplete data as obtained from a classifier's output. Wireless data is noisy, resulting in the capture of incomplete data properties, as [63] demonstrates. Thus, when there is only a small amount of classification data available, generation of the stolen model can be aided by human judgement. In [63], the authors reconstructed a model by which to predict lung cancer from pulmonary data. The captured data was displayed as 3D images, allowing humans to add certain properties, such as marking of lesions, with this ensuring the creation of labelled input data. With this data, the authors developed a stolen model, employing a Convolutional Neural Network. The technique showed that the stolen and original model differed by 0.3% in terms of accuracy.

Tab. III. summarises the discussion on model stealing. The challenges here can be viewed from the location of the machine learning-based classifier that transforms data, i.e. either at the edge, in the Cloud, or on a mobile device. The issue is based fundamentally upon technique for data acquisition. The more remote the classifier is, the more limited the data acquired. Thus, AI techniques play a crucial role in estimating the data parameters to help an adversary achieve its goals.

2.3.2. Discussion

The above discussion shows that model stealing techniques collect, estimate and create data. It is therefore possible for data to be collected into a rich database, oracle or knowledgebase (as the authors of [33] demonstrated), to provide data that facilitates future model stealing-based adversarial activities. As Fig. 5. illustrates,

when such a knowledgebase also incorporates solutions from the evasion techniques, an adversarial bot or malware can act rationally with the aim of carrying out cyberattacks. If the rational bots or malware knows the victim's model, it can select the best tool for reconnaissance, discretely open ports and replicate itself, maintain persistence on the target system, escalate privilege, and conduct lateral movement to attack other platforms. An AI agent that is equipped with both evasion and model-stealing techniques can mimic the normal data parameters when pursuing the above attack chain, creating undetectable attacks.

2.4. Solutions to mitigate adversarial AI attacks

While the previous subsection formulates AI-based cyberattacks, this subsection discusses their mitigation techniques to defend against adversarial AI attacks. Principally, these methods derive from, first, the assumption that the victims had significantly more knowledge about their own network/system parameters than their adversaries. Second, that the adversarial techniques would have some disadvantages inherent to them. Thus, the mitigation techniques leverage these adversarial techniques' disadvantages.

2.4.1. Feature definition

A set of features is what enables a machine-learning technique to engage in classification. When an adversary has obtained the feature set, they can mimic the way the machine learning technique classifies data. In this case, the defenders' option would be to redefine the set of features for data analysis. This is made possible when, for example, a new technology is introduced, causing the defender to re-analyse data. In [34], the authors showed that, when the new web communications protocol HTTP/2 was introduced, the traffic pattern was different from its predecessor, HTTP/1.1. This allowed adversaries to create attack traffic undetectable by machine-learning techniques. Thus, the authors of [34] proposed a new set of features to allow for the detection of the stealthy attack traffic.

The defining of a new set of features follows the data mining technique, such as the one described in [64]. A feature is identified either by observation or intuition. A good feature has a value distribution that can identify the intended class (e.g. attack or legitimate) closely. For example, a feature with wide data distribution can allow a threshold to be placed for classification (Fig. 6). When data is multidimensional, a set of identified features is ranked according to how well the combination can lead to a classification. Algorithms such as information gain [65] are used commonly in ranking features.

Both the adversary and the defender can find a set of features for attack modelling as well as for mitigation, as in the case of Cloud services. Cloud services see incoming requests from any client connecting to them, of either legitimate or fake status. When a machine-learning service is deployed in the Cloud, it is vulnerable to fake queries that can be crafted for the purpose of model stealing. That is, from the adversary perspective, the distribution of the input dataset can be inferred (as discussed in Section 2.3), allowing adversaries to define morphed features. To detect adversarial query traffic from the defender's perspective, the authors in [62] analysed the distribution difference between legitimate and adversarial query traffic. They then proposed a set of features by which to detect the adversarial query traffic, allowing them to detect abnormalities in the input data with 92% accuracies.

2.4.2. Monitoring

As discussed in [66], network management involves configuration and measurement. Monitoring is the activity of collecting and analysing network measurements, which depict the network's behaviour and performance. Monitoring involves the collection, analysis, and presentation of data (Fig. 8). The collection layer captures

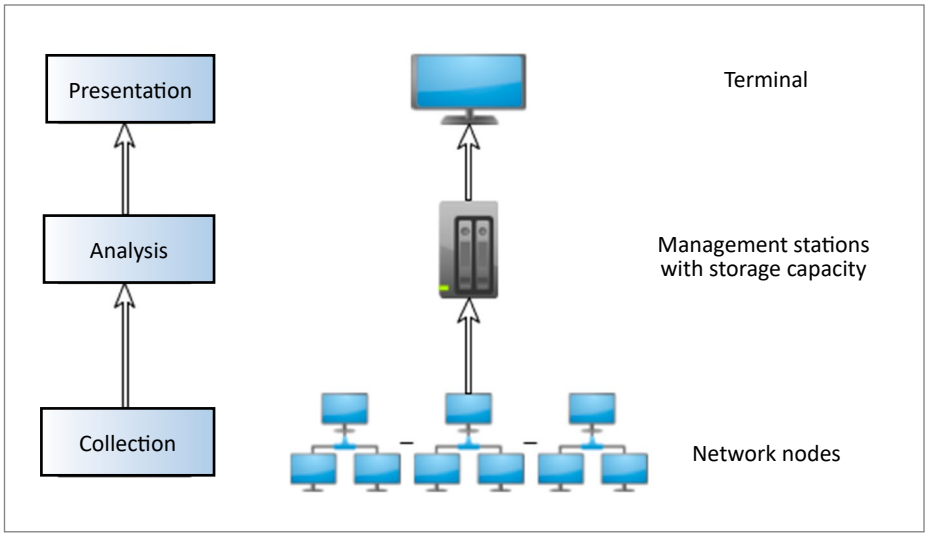

Figure 8. Monitoring as the management layer in measuring network performance, adapted from [66].

network traffic; the analysis layer extracts network traffic and converts the same into relevant data; and the presentation layer provides for meaningful (e.g. graphic) representation of the data that interprets network behaviour and performance. Because such network monitoring is sourced from network traffic generated from the whole network under management, cyber-network owners have better knowledge of their own network than the adversary.

In [40], the authors observed that the adversaries can only access a subset of adversarial nodes situated in the network to generate stealthy attack traffic. A compromised node can manipulate only a subset of its neighbouring nodes to believe (i.e. measure) that certain packets have been forwarded. However, not all nodes can be manipulated in a single broadcast, because traffic-volume configuration and historical measurement differ from one node to another. Thus, the authors proposed to expand the monitoring of the network so that more neighbouring nodes count the number of packets forwarded from each node. If one node has differing belief than another as to the number of packets being forwarded by a monitored node, then packet dropping had occurred.

Similarly, in critical infrastructure networks, the authors of [41] observed that attackers face non-negligible risks of being detected if they generate attack traffic imprudently. This is because the behaviour of the network can become unpredictable to the attacker. In simulating attacks, the author of [41] assumed that the attackers know the distribution of the input sensor values (with consumer behaviour considered to consume utility over time). However, sensor values depend on complex interactions with other sensors in real network settings. Attackers have less knowledge than defenders in predicting consumer behaviour and random perturbations in the network. Thus, the deploying of an extensive monitoring system throughout the network can confer advantage upon the defender.

2.4.3. External validation

External validation offers an innate defence against adversarial AI, because the efficacy of research findings may not be as valid when applied to real settings. Research conducted by the authors of [41], for example, simulated the critical infrastructure out of a room-sized lab, showing how pH levels in water can be manipulated. The infrastructure consisted of 6 main Programmable Logic Controllers (PLCs), one of which was to control the pH level. The study [41] assumed that a man-in-the-middle attack succeeded in controlling the PLC for pH level, causing the water

acidity to change linearly, such that $pH_{time+1} = pH_{time} + delay$. Such lab-based behaviour is more predictable than that in real settings, where there would be more PLCs, constraints placed on how long a PLC can be turned on/off, and complex filtering systems for neutralizing water parameters. This would all ensure the potential failure of an attack under real settings.

In [60], the authors recognised that the solutions in model stealing are impractical in reconstructing real-world image classifications. The problems are that the input dataset (Fig. 7) is not readily available to the attackers. Attackers only assume partial availability of the data, or the availability of a similar dataset. As such, attacks become ineffective when deployed in real settings. Furthermore, the authors in [60] discussed the fact that, while GANs were mostly used to construct a stolen model, this is not ideally applied to real image classification settings because the dimensionality of the generator's parameters can be in the order of millions.

Recall from Section 2.2.2. that image recognition can play roles in protecting cyber-infrastructure. In [43], a study showed how to evade a machine-learning technique that is recognising a person's face. The study used data from images taken with room lighting without exterior windows to prevent extreme lighting variations. The persons whose faces were taken as data samples must maintain a neutral expression and were stood at a predetermined distance from the camera. Such data are not valid externally; the authors acknowledged that detection using real outdoor images is challenging. This means that one defensive method against adversarial AI attacks is to embrace the degrading external validity, for example by creating a more complex system.

2.4.4. Alteration of parameters

As Fig. 6 shows, adversaries can infer a classifier's parameters, such as the threshold, to mimic the distribution of target data. The approach to defending against such a scenario is thus to change parameters, either as the combination of parameters used by the classifier or as parameter values. In [44], the adversaries inferred both the threshold and the window size of the target data, allowing them to inject false data to change a drone's position. However, when these parameters were changed as part of a defensive method, the room for manoeuvre to inject false data was decreased. This is illustrated in Fig. 9. Under a normal situation, a drone's position does not deviate from the allowed position threshold. When false data were injected, however, the position deviated but was still below the threshold (undetected attack 1, Fig. 9.). Even though a drone's position deviated above the threshold, this did not raise an alarm because the event was construed as just a fraction of the time window (undetected attack 2, Fig. 9.). To mitigate these attacks, the authors of [44] suggested having an adaptive threshold and variable-size window which will result in the impact of the attack being diminished. In an adaptive threshold (mitigation 1, Fig. 9.), the allowed position deviation threshold was changed, causing the false data to fall above it. In a variable-size window (mitigation 2, Fig. 9.), the injected false data still reached above the threshold at all times, precisely because the window size was reduced.

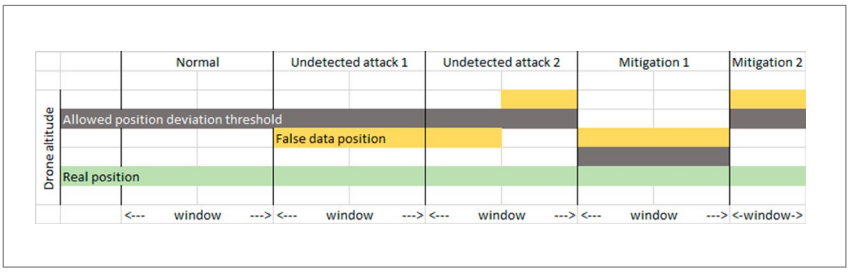

Figure 9. To detect anomalies a defensive method can alter either the threshold or the window size.

The second method is to change the parameter values of the machine-learning model relevant for classification prediction to a higher level of precision. As the authors of [58] have shown, Cloud machine learning services round their values to two decimals to provide only the necessary information. Yet, Cloud services can make internal use of the higher-precision values to generate their output data. By supplying only a fraction of the data at the output, i.e., rounded values to a lower precision, Cloud services can defend against model-stealing attacks.

2.4.5. Adversarial training

In adversarial training [67], adversarial attack techniques are used to train the system to be defended. The technique uses perturbed input data, which represent the attack samples, to cause machine-learning models to misclassify. Perturbing input data is crafting data samples such that their feature values are modified by a small deviation from their original value, causing the machine-learning algorithm to create a deviated / wrong function, to be used as the model. Collectively, the deviated values maximise the loss between the intended function and the modified function. Adversarial training is to generate a sizeable, perturbed input dataset, label the correct class, and use such perturbed input data as the training data to train the machine-learning model. Thus, the system becomes more secure because attack samples have been seen during the training phase. The technique is considered the accepted procedure by which to defend against the perturbation attacks discussed in the previous section. In this case, the system has seen perturbed inputs.

Traditionally, adversarial training was used for image classification in computer vision [67]. The technique is now adopted in sensitive domains such as cyber-physical systems [45], critical infrastructures [46], and industrial-control systems [47]. With adversarial training, the training data can be perturbed to represent the complexity of the physical system, e.g. in relation to actuator constraints in water systems [45]; voltage, current data, short-circuit fault, line maintenance, remote tripping, and relay settings in a power grid [46, 47].

2.4.6. Patching software

A patching of software vulnerability represents an effective approach by which to defend against fuzzing attack. Conceptually, this is like the adversarial training described above; with software being fuzzed to find, and eventually patch, vulnerability. There are two advantages to the patching of software. First, non-limitation solely to the securing of vulnerability, but also an increasing of complexity in the software execution path. The disadvantage of fuzzing is that it relies on code coverage [49]. This is to say that, the larger the code, the less successful is the fuzzing attack. Thus, fuzzing would not discover all the execution paths of the software following software patching. Second, the authors of [50] point out that the application of different AI techniques in fuzzing the same problem can lead to significant differences in the discovery of execution paths for attacks. An increases in the number of execution paths as software is patched can lead to a lowering of the success rate where fuzzing attacks are concerned.

2.4.7. Defensive distillation

Rather than having a machine-learning model that outputs a class with a high probability, the defensive distillation technique [59] suggests that the model smooths the probability of the output class. This causes the probability of the model generating one class to be similar to the probability that the model generates the other classes. The technique is thus suitable when it comes to defending against model-stealing attacks. One variation of the technique [58] is for the classifier to output only the top-n classes with the highest probabilities. This would limit the adversary's knowledge of what classes would have a low probability with a given sample. The technique can be enhanced further where only the top-1 class is outputted. Although in

[58] the authors observe that there was no significant advantage in defending against model stealing attacks using the top-1 defence, they still believe this to be the logical defence technique, given the way it provides users with very limited information.

3. Future directions

The concept of AI-based cyberattacks has emerged from the convergence of AI algorithms, a rapid increase in computational power, application development and operation advancements, and the ready availability of AI-based implementations for adversarial adoption.

Future AI-based attacks are also determined by attacker motives. An increasingly significant threat is posed by lucrative opportunities for the adversary through cyber threats posed in contemporary times that involve hijacking of systems and encryption of user data, even as the latter are held for ransom (i.e. through ransomware), with user payments in cryptocurrency form demanded. Such opportunities may emerge through AI-based attacks that assess the vulnerabilities of victim machines, in advance of their sending a ransomware payload through to them. Motives other than financial gain can include terrorism, business competition (e.g. as bots spread fake news), hacktivism or the expression of political views. The adversarial AI techniques discussed in this work (Section 2.1.) can be used to achieve such motives.

As we have discussed through the analysis reported in this contribution, the ability of computing platforms to thwart the AI-based cyberattack spectrum depends on the following observed aspects:

1. the design of computing platforms resilient to AI-based adversarial threats, not as an afterthought to production, but rather via an awareness that everybody has responsibility (as part of the DevSecOps paradigm);
2. the design of applications (web-based, mobile, and Cloud) that are resilient to AI-based cyber-threats, by way of the prevention of data capturing and fostering of attacker learning through the provisioning of feedback, i.e. a reduced amount of feedback data provisioned to end-users given the possibility of comprising both legitimate and adversary class;
3. the design of network security controls adopted in a network, with a view to cyber-threats arising through AI engine exploitation being thwarted (future directions for such activity may include egress and ingress packet-filtering based on detection of anomalous feedback patterns (statistical as well as pattern-based) that are moving through the network;
4. the identification of opportunities to obfuscate-neural network operations, parameters, and generated outputs, and the adoption of a black box-based framework by which to prevent the adversary from exploiting system weakness in provisioning of clearly indicative data to the adversary;
5. adoption as common practice of security by design, even as heterogeneity is incorporated in the nature and type of IT and Operations Technology (OT) devices that comprise a modern-day Information Communication Technology (ICT) platform (i.e. IoT and edge devices, digital controllers, Supervisory Control and Data Acquisition (SCADA) systems, Cloud servers and mobile devices) - design should include options to prevent AI-based cyber threats from being perpetrated against the holistic platform identified above.

Future adversarial AI attacks will become more pervasive over time. In modern Internet settings, the generation, exchange and processing of data rely on remote data operations, including those found in the evolving discipline of Industrial IoT (IIoT). As we have discussed in Section 1, each data- exchange end point on the Internet is vulnerable to exploitation. Modern society has become more dependent on the integrity and availability of such cyber-services as are seen in banking and critical infrastructures. For example, the integration of IoT devices with back-end Clouds is

a common practice in contemporary computing domains including critical infrastructures, as where the aims are electricity distribution, load balancing, and feed-in tariff in smart grids. The ability of the adversary to determine IT/OT vulnerabilities within a smart grid may prove catastrophic to the routine operations of critical infrastructure, which is essential to provision routine services such as electricity distribution to citizens.

Another emerging field of study is the digital forensic readiness of AI-based systems. Essentially, the vulnerabilities of such systems can be exploited by the adversary through the adoption of technologically-advanced tactics, including the circumvention of facial recognition systems, the bypassing of the security controls of AI systems, and the deliberate injection of falsified data into the communication stream. By analysing such empirical data, it is possible to hoard the right data types and data artefacts as may help a digital forensic investigation undertaken as part of post-incident analysis.

4. Conclusion

AI techniques have gained use, not only to defend traditional network systems, but also to attack their implementations. This is made possible because the modern Internet is exchanging not only raw data, but also processed data such as that generated by Cloud-based machine-learning services. This phenomenon is seen to affect the cyber infrastructure comprising enterprise, mobile, and autonomous systems, as these engage in the exchange of both sensory data and AI analytical data. Such infrastructure has become the playground for AI-based cyberattacks. The number of possibilities to attack – from reconnaissance, execution, persistence, privilege escalation, command/control, to data exfiltration-manifests as too large a spectrum for humans to analyse zero-day attacks. However, adversaries are one step ahead, using knowledgebase and known AI techniques to launch AI-based cyberattacks.

AI techniques are suitable for defining attack vectors because they can handle large volumes of data. In this paper, we described machine-learning-based techniques for adversarial AI. These techniques are adopted by the adversary to carry out adversarial AI attacks, with the derivatives of neural networks playing a significant role in fostering the development of novel AI attack vectors incorporating both spatial and temporal data in the emulation of legitimate data. We further classified Adversarial AI into behaviour mimicry (which employs stealthy attacks, perturbation and fuzzing techniques) and rational bot (which employs the model-stealing technique). Behaviour-mimicry techniques aim to resemble normal data, thus infiltrating victim machines. Rational bots employ the knowledgebase that was obtained from the model-stealing technique to facilitate the design of zero-day attack vectors.

Also discussed here are 7 methods by which to defend against AI-based cyberattacks. The mitigation approaches leverage the disadvantages understood from the adversarial AI techniques. And, even in the face of these countermeasures, adversarial AI attacks will be – as we indicate – more pervasive, as society becomes more dependent on cyber-data exchanges that offer a plethora of opportunities for adversaries to further their motives effectively.

Acknowledgments

We thank the anonymous reviewers for their valuable comments of assistance to us in improving the content, organisation, and presentation of this paper. Sherali Zeadally was supported by a Fulbright U.S. Scholar Grant Award administered by the U.S. Department of State's Bureau of Educational and Cultural Affairs, and through its cooperating agency the Institute of International Education ("IIE").

REFERENCES

[1] I. Novikov. (2018). *How AI Can Be Applied To Cyberattacks* [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2018/03/22/how-ai-can-be-applied-to-cyberattacks/#3211152d49e3. [Accessed: Sep. 28, 2022].

[2] S. Zeadally, E. Adi, Z. Baig, I. A. Khan, "Harnessing artificial intelligence capabilities to improve cybersecurity," *IEEE Access*, vol. 8, pp. 23817–23837, 2020, doi: 10.1109/ACCESS.2020.2968045.

[3] J. Hobbs. (2018). *AI Enters the Cyber Attack Realm* [Online]. Available: https://www.afcea.org/content/ai-enters-cyber-attack-realm. [Accessed: Sep. 28, 2022].

[4] E. Adi, A. Anwar, Z. Baig, S. Zeadally, "Machine Learning and Data Analytics for the IoT," *Neural Computing & Application*, vol. 32, pp. 16205–16233, 2020, doi: 10.1007/s00521-020-04874-y.

[5] S. Russell, P. Norvig, *Artificial intelligence: a modern approach*, no. 4London: Pearson Education, 2020.

[6] MITRE. *MITRE ATT&CK* [Online]. Available: https://attack.mitre.org/matrices/. [Accessed: Sep. 28, 2022].

[7] *Lockheed Martin, The Cyber Kill Chain* [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html. [Accessed: Sep. 28, 2022].

[8] I. Corona, G. Giacinto, F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013, doi: 10.1016/j.ins.2013.03.022.

[9] M. Babar, M. Sohail Khan, "ScalEdge: A framework for scalable edge computing in internet of things–based smart systems," *International Journal of Distributed Sensor Networks*, vol. 17, no. 7, pp. 1–11, 2021, doi: 10.1177/155014772110353.

[10] J. Neeli, S. Patil, "Insight to security paradigm, research trend & statistics in internet of things (iot)," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 84–90, 2021, doi: 10.1016/j.gltp.2021.01.012.

[11] I. Rosenberg, A. Shabtai, Y. Elovici, L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021, doi: 10.1145/3453158.

[12] A. McCarthy, E. Ghadafi, P. Andriotis, P. Legg, "Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey," *Journal of Cybersecurity and Privacy*, vol. 2, no.1, pp. 154–190, 2022, doi: 10.3390/jcp2010010.

[13] E. Alhajjar, P. Maxwell, N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications*, vol. 186, pp. 1–25, 2021, doi: 10.48550/arXiv.2004.11898.

[14] H. Navidan, P. F. Moshiri, M. Nabati, R. Shahbazian, S. A. Ghorashi, "Generative adversarial networks (gans) in networking: A comprehensive survey & evaluation," *Computer Networks*, vol. 194, no. 3, pp. 1–26, 2021, doi: 10.1016/j.comne vol.2021.108149.

[15] D. Li, Q. Li, Y. Ye, S. Xu, "Arms race in adversarial malware detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–35, 2021, doi: 10.1145/3484491.

[16] M. Pawlicki, M. Choraś, R. Kozik, "Defending network intrusion detection systems against adversarial evasion attacks," *Future Generation Computer Systems*, vol. 110, pp. 148–154, 2020, doi: 10.1016/j.future.2020.04.013.

[17] Z. Guan, L. Bian, VOL. Shang, J. Liu, "When Machine Learning meets Security Issues: A survey," *2018 International Conference on Intelligence and Safety for Robotics*, Shenyang, 2018, pp. 158–165 [Online]. Available: https://ieeexplore.ieee.org/document/8535799. [Accessed: Sep. 28, 2022].

[18] G. Apruzzese, M. Colajanni, L. Ferretti, M. Marchetti. (2019). "Addressing Adversarial Attacks Against Security Systems Based on Machine Learning," *11th International Conference on Cyber Conflict (CyCon)*, pp. 1–18 [Online]. Available: https://ccdcoe.org/uploads/2019/06/Art_21_Addressing-Adversalial-Attacks.pdf. [Accessed: Sep. 28, 2022].

[19] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, J. D. Tygar, "Adversarial machine learning," *In Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.

[20] V. Duddu, "A Survey of Adversarial Machine Learning in Cyber Warfare," *Defence Science Journal*, vol. 68, no. 4, pp. 356–366, 2018, doi: 10.14429/dsj.68.12371.

[21] A. Szychter, H. Ameur, A. Kung, D. Hervé. (2018). *The Impact of Artificial Intelligence on Security: a Dual Perspective* [Online]. Available: https://www.cesar-conference.org/wp-content/uploads/2018/11/articles/C&ESAR_2018_J1-03_A-SZYCHTER_Dual_perspective_AI_in_Cybersecurity.pdf. [Accessed: Sep. 28, 2022].

[22] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," *Nature*, vol. 521, p. 436–444, 2015, doi: 10.1038/nature14539.

[23] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, *MIT press*, 2016.

[24] M. M. Gamal, B. Hasan, A. F. Hegazy, "A security analysis framework powered by an expert system," *International Journal of Computer Science and Security (IJCSS)*, vol. 4, no. 6, pp. 505–527, 2011.

[25] J. Kennedy, R. Eberhart. (1995). "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 [Online]. Available: https://ieeexplore.ieee.org/document/488968. [Accessed: Sep. 28, 2022].

[26] M. H. Nasir, S. A. Khan, M. M. Khan, M. Fatima, "Swarm intelligence inspired intrusion detection systems—a systematic literature review," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 205, pp. 108708, 2022, doi: 10.1016/j.comnet.2021.108708.

[27] VOL. Bayes, LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, In a letter to John Canton, AMFR S, Philosophical transactions of the Royal Society of London, vol. 53, pp. 370–418, 1763, doi: 10.1098/rstl.1763.0053.

[28] C. Cortes, V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[29] W. S. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, vol. 4, pp. 115–133, 1943, doi: 10.1007/bf02478259.

[30] I. H. Sarker, "Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective," *SN Computer Science*, vol. 2, no. 3, pp. 1–16, 2021, doi: 10.1007/s42979-021-00535-6.

[31] Z. Fang, J. Wang, B. Li, S. Wu, Y. Zhou, et.al., "Evading Anti-Malware Engines with Deep Reinforcement Learning," *IEEE Access 7*, 2019, pp. 48867–48879, 2019. doi: 10.1109/ACCESS.2019.2908033.24.

[32] S. Sen, E. Aydogan, A. I. Aysan, "Coevolution of Mobile Malware and Anti-Malware," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2563–2574, 2018, doi: 10.1109/TIFS.2018.2824250.

[33] E. Zouave, M. Bruce, K. Colde, M. Jaitner, I. Rodhe, "*Artificially intelligent cyberattacks*", Stockholm: Totalförsvarets forskningsinstitut FOI [Online] Available: https://whttps://www.statsvet.uu.se/digitalAssets/769/c_769530-l_3-k_rapport-foi-vt20.pdf [Accessed: Sep.28, 2022].

[34] E. Adi, Z. Baig, P. Hingston, "Stealthy Denial of Service (DoS) attack modelling and detection for HTTP/2 services," *Journal of Network and Computer Applications*, vol. 91, pp. 1–13, 2017, doi: 10.1016/j.jnca.2017.04.015.

[35] A. M. Turing, "Computing machinery and intelligence," in *Parsing the turing test*, doi: 10.1007/978-1-4020-6710-5_3.

[36] MITRE. *Supply Chain Compromise for Enterprise [Online]. Available:* https://attack.mitre.org/techniques/T1195/. [Accessed: Sep. 28, 2022].

[37] MITRE. *Supply Chain Compromise for Mobile* [Online]. Available: https://attack.mitre.org/techniques/T1474/. [Accessed: Sep. 28, 2022].

[38] MITRE. *Supply Chain Compromise for Industrial Control System* [Online]. Available: https://collaborate.mitre.org/attackics/index.php/Technique/T0862. [Accessed: Sep. 28, 2022].

[39] E. Fix, J. L. Hodges, "Discriminatory analysis-nonparametric discrimination: Consistency properties," *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989, doi: 10.2307/1403797.

[40] I. Khalil, S. Bagchi, "Stealthy attacks in wireless ad hoc networks: Detection and countermeasure,", *IEEE Transactions on Mobile Computing*, vol. 10, no. 8, pp. 1096–1112, 2011, doi: 10.1109/TMC.2010.249.

[41] D. I. Urbina, J. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, et al., "Limiting the impact of stealthy attacks on Industrial Control Systems," *Proceedings of the ACM Conference on Computer and Communications Security*, 2016, pp. 1092–1105 [Online], Avaiable https://users.soe.ucsc.edu/~alacarde/papers/ccs16.pdf. [Accessed: Sep. 28, 2022].

[42] B. Filkins, D.Wylie, A. Dely, "Sans 2019 state of ot/ics cybersecurity survey," SANS Institute, 2019.

[43] M. Sharif, S. Bhagavatula, L. Bauer, M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," *Proceedings of the ACM Conference on Computer and Communications Security*, 2016, pp. 1528–1540 [Online]. Available: https://users.ece.cmu.edu/~lbauer/papers/2016/ccs2016-face-recognition.pdf. [Accessed: Sep. 28, 2022].

[44] P. Dash, M. Karimibiuki, K. Pattabiraman, "Out of control: Stealthy Attacks against Robotic Vehicles Protected by Control-based Techniques," *ACM International Conference Proceeding Series*, 2019. pp. 660–672.

[45] J. Li, Y. Yang, J. S. Sun, K. Tomsovic, H. Qi, "Conaml: Constrained adversarial machine learning for cyber-physical systems," *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 52–66 [Online]. Available: https://par.nsf.gov/servlets/purl/10314482. [Accessed: Sep. 28, 2022].

[46] I. Niazazari, H. Livani, "Attack on Grid Event Cause Analysis: An Adversarial Machine Learning Approach," 2019, doi: 10.48550/arxiv.1911.08011.

[47] E. Anthi, L. Williams, M. Rhode, P. Burnap, A. Wedgbury, "Adversarial attacks on machine learning cybersecurity defences in industrial control systems," *Journal of Information Security and Applications*, vol. 58, no. 8, pp. 102717, 2021, doi: 10.1016/j.jisa.2020.102717.

[48] M. Rajpal, W. Blum, R. Singh, "Not all bytes are equal: Neural byte sieve for fuzzing," *arXiv preprint arXiv:1711.04596*, pp. 1–10, 2017.

[49] J. Li, B. Zhao, C. Zhang, "Fuzzing: a survey," *Cybersecurity*, vol. 1, pp. 1–13, 2018, doi: 10.1186/s42400-018-0002-y.

[50] Y. Wang, P. Jia, L. Liu, C. Huang, Z. Liu, "A systematic review of fuzzing based on machine learning techniques," *PLoS ONE*, vol. 15, no. 8, pp. 1–37, 2020, doi: 10.1371/journal.pone.0237749.

[51] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, et al., "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 19–35 [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8418594. [Accessed: Sep. 28, 2022].

[52] D. Yacchirema, J. S. de Puga, C. Palau, M. Esteve, "Fall detection system for elderly people using IoT and ensemble machine learning algorithm," *Personal and Ubiquitous Computing*, vol. 23, no. 5-6, pp. 801–817, 2019, doi: 10.1007/s00779-018-01196-8.

[53] T. Takahashi, Y. Kadobayashi, "Reference ontology for cybersecurity operational information," *The Computer Journal*, vol. 58, no. 10, pp. 2297–2312, 2015, doi: 10.1093/comjnl/bxu101.

[54] MITRE. *Lateral Movement for Enterprise* [Online]. Available: https://attack.mitre.org/tactics/TA0008/. [Accessed: Sep. 28, 2022].

[55] MITRE. *Lateral Movement for Mobile* [Online]. Available: https://attack.mitre.org/tactics/TA0033/26. [Accessed: Sep. 28, 2022].

[56] MITRE. *Lateral Movement for Industrial Control Systems* [Online]. Available: https://collaborate.mitre.org/attackics/index.php/Lateral_Movement. [Accessed: Sep. 28, 2022].

[57] M. Chorań, M. Pawlicki, R. Kozik, "The feasibility of deep learning use for adversarial model extraction in the cybersecurity domain," *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 2019, pp. 353–360.

[58] X. Yuan, L. Ding, L. Zhang, X. Li, D. Wu, "Es attack: Model stealing against deep neural networks without data hurdles," *arXiv preprint arXiv:2009.09560*, 2020.

[59] G. Hinton, O. Vinyals, J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[60] S. Kariyappa, A. Prakash, M. K. Qureshi, "Maze: Data-free model stealing attack using zeroth-order gradient estimation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13814–13823.

[61] S. Ghadimi, G. Lan, "Stochastic first- and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013, doi: 10.1137/120880811.

[62] H. Yu, K. Yang, T. Zhang, Y.-Y. Tsai, T.-Y. Ho, et al., "Cloudleak: Large-scale deep learning models stealing through adversarial examples," *NDSS*, 2020. doi: 10.14722/ndss.2020.24178.

[63] L. Zhang, G. Lin, B. Gao, Z. Qin, Y. Tai, J. Zhang, "Neural model stealing attack to smart mobile device on intelligent medical platform," *Wireless Communications and Mobile Computing*, vol. 2020, doi: 10.1155/2020/8859489.

[64] I. H. Witten, E. Frank, M. A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques," *4th Edition, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann*, Amsterdam, 2017 [Online]. Available: http://www.sciencedirect.com/science/book/9780123748560. [Accessed: Sep. 28, 2022].

[65] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1023/A:1022643204877.

[66] S. Lee, K. Levanti, H. S. Kim, "Network monitoring: Present and future," *Computer Networks*, vol. 65, pp. 84–98, 2014, doi: 10.1016/j.comnet.2014.03.007.

[67] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, et al., "Ensemble adversarial training: Attacks and defenses," 6th *International Conference on Learning Representations*, 2018, pp. 1–20 [Online]. Available: https://floriantramer.com/docs/papers/iclr18ensemble.pdf. [Accessed: Sep. 28, 2022].