2020 12th International Conference on Cyber Conflict
20/20 Vision: The Next Decade
T. Jančárková, L. Lindström,
M. Signoretti, I. Tolga, G. Visky (Eds.)
2020 © NATO CCDCOE Publications, Tallinn

Hacking the AI - the Next Generation of Hijacked Systems

Kim Hartmann
Conflict Studies Research Centre
kim.hartmann@conflictstudies.org.uk

Christoph Steup
Anhalt University of Applied Sciences
christoph.steup@hs-anhalt.de

Abstract: Within the next decade, the need for automation, intelligent data handling and pre-processing is expected to increase in order to cope with the vast amount of information generated by a heavily connected and digitalised world. Over the past decades, modern computer networks, infrastructures and digital devices have grown in both complexity and interconnectivity. Cyber security personnel protecting these assets have been confronted with increasing attack surfaces and advancing attack patterns. In order to manage this, cyber defence methods began to rely on automation and (artificial) intelligence supporting the work of humans. However, machine learning (ML) and artificial intelligence (AI) supported methods have not only been integrated in network monitoring and endpoint security products but are almost omnipresent in any application involving constant monitoring, complex or large volumes of data. Intelligent IDS, automated cyber defence, network monitoring and surveillance as well as secure software development and orchestration are all examples of assets that are reliant on ML and automation. These applications are of considerable interest to malicious actors due to their importance to society. Furthermore, ML and AI methods are also used in audio-visual systems utilised by digital assistants, autonomous vehicles, face-recognition applications and many others. Successful attack vectors targeting the AI of audio-visual systems have already been reported. These attacks range from requiring little technical knowledge to complex attacks hijacking the underlying AI.

With the increasing dependence of society on ML and AI, we must prepare for the next generation of cyber attacks being directed against these areas. Attacking a system through its learning and automation methods allows attackers to severely damage the system, while at the same time allowing them to operate covertly. The combination

of being inherently hidden through the manipulation made, its devastating impact and the wide unawareness of AI and ML vulnerabilities make attack vectors against AI and ML highly favourable for malicious operators. Furthermore, AI systems tend to be difficult to analyse post-incident as well as to monitor during operations. Discriminating a compromised from an uncompromised AI in real-time is still considered difficult.

In this paper, we report on the state of the art of attack patterns directed against AI and ML methods. We derive and discuss the attack surface of prominent learning mechanisms utilised in AI systems. We conclude with an analysis of the implications of AI and ML attacks for the next decade of cyber conflicts as well as mitigations strategies and their limitations.

Keywords: *AI hijacking, artificial intelligence, machine learning, cyber attack, cyber security*

1. INTRODUCTION

Artificial intelligence (AI) has been applied in many scenarios in recent years, and this technology is expected to establish itself in further fields over the next decade. Within the military sphere alone, AI technology is expected to penetrate into areas such as intelligence, surveillance, reconnaissance, logistics, cyberspace operations, information operations (the most prominent technology is currently "deepfakes"), command and control, semiautonomous and autonomous vehicles and autonomous weapon systems. Numerous reports and analyses suggest that an AI arms race has indeed already begun [1]. In addition to the military application scenarios, AI systems are also utilised in applications such as public security surveillance [2], financial markets [3], healthcare [4], Human-Computer and Human-Machine Interactions, cybersecurity, power grid management [5], autonomous driving and driver assistance systems. Any of the aforementioned application scenarios are of high value to civilian, governmental or military units and have a high significance to society. Therefore, these applications and the systems involved must be considered as highly valuable assets in cyberwarfare and protected accordingly.

The security of AI systems is currently underrepresented in public discussions; however, reports on successful attacks on AI systems have emerged over the past couple of years. The utilised attack vectors range from requiring little technical

expertise to attacks involving detailed knowledge of the underlying AI [6]. Reported results have ranged from the AI mistaking a turtle for a rifle, to making individuals undetectable to the system.

The penetration of AI throughout digital spaces is likely to increase even further over the next decade, as well as our reliance on its correct identification and reasoning abilities. AI is envisioned to outperform humans in most tasks involving processing large amounts of data/information, high precision or complex reasoning. It is assumed to deliver unbiased and rational results without interference from non-logical events or circumstances. This presumption renders hijacked AI systems an extremely dangerous threat to modern societies.

The wide-range of applications involving AI is startling, especially as AI has been regarded as being almost impossible to secure [7]. In December 2019, Microsoft published a series of materials on the topic, stating that "[i]n short, there is no common terminology today to discuss security threats to these systems and methods to mitigate them, and we hope these new materials will provide baseline language […]" [8]. Over the past decade, we have witnessed increasing and incautious utilisation of AI and ML techniques in applications whose correct functioning is crucial to modern societies. It is easy to imagine how any malfunctioning of these systems might have a devastating impact on civilian lives, financial markets, national security and even military operations.

With society's increasing dependence on ML and AI, we must prepare for the next generation of cyber attacks being directed against these systems. Attacking the system through its learning and automation methods allows the attackers to severely damage the system by altering its learning outcome, decision making, identification or final output. Furthermore, it is difficult to analyse AI systems post-incident and integrate real-time monitoring during their operation: much of the learning and reasoning is done in what is called a "hidden layer" and in its essence corresponding to a black box model. Therefore, the discrimination of a compromised from an uncompromised AI system in real-time is still considered very difficult. With its increasing utilisation in crucial application scenarios, the security of AI systems becomes indispensable.

Knowledge of AI systems' vulnerabilities may also become of high importance to defensive cyber operations. During 2019, we witnessed increasing weaponisation of AI, often to create "deepfakes" – artificially generated or altered media material found to impose a sincere threat to democracies [9]. The uprising of deepfakes has encouraged the U.S. DARPA to spend $68 million on the identification of deepfakes over the past four years [10]. While it is of utmost importance to identify AI-supported disinformation campaigns, identification alone will not stop such operations. Offensive

technological knowledge of how to stop AI-supported attacks will become essential to establish and uphold cyber power in an ongoing AI arms race.

The aim of this paper is to foster understanding of the susceptibility of AI systems to cyber attacks, how incautious utilisation of AI and ML may make societies vulnerable, and to transfer the value of knowing AI-/ML-system vulnerabilities within the ongoing AI arms race. Attack surface modelling is a key contribution to assessing a target's susceptibility to attacks. However, AI systems have several peculiarities, which must be addressed when deriving the attack surface. Within this article, attack surfaces of different AI systems are derived that consider systems' data assets, processing units and known attack vectors, allowing us to understand these systems' vulnerabilities. Furthermore, these attack surfaces must be discussed with the systems' societal and economic impact in mind to allow strategic and policy recommendations. At the time of writing, neither the AI systems' concrete attack surface definition nor the embedment of the different AI systems' specific operational setup have been part of the security assessment of these systems. Allowing an AI-specific, concrete attack surface discussion, which includes the operational setup associated with the AI/ML method utilised by the system, is the main contribution of this article in addition to providing insights into the role of AI systems' susceptibilities to cyber attacks in the next decade of cyber conflicts.

This paper will continue as follows: we start by giving a brief introduction to selected AI and ML methods currently deployed (section 2). We report on state of the art attack patterns directed against these systems and how it must be expected that these systems will become prominent targets over the next decade. We derive and discuss how attack surfaces may be modelled for AI systems (section 3). In section 4, we apply the previously derived attack surface model to AI systems utilising the different methods previously introduced in section 2 to compare their susceptibility to attacks. We conclude with an analysis of the implications of AI and ML attacks for the next generation of cyber conflicts and recent mitigation strategy attempts (section 5).

2. AI AND ML METHODS

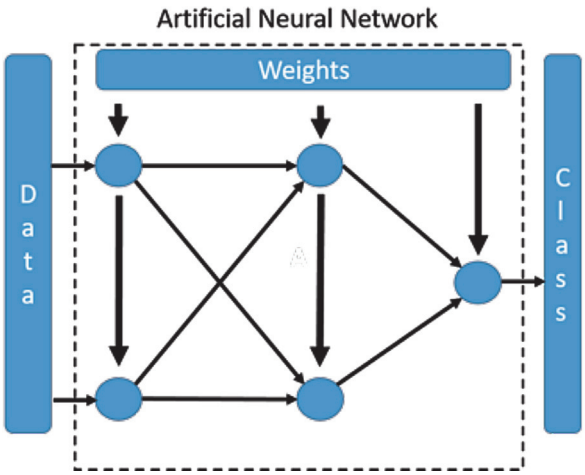
The field of artificial intelligence and especially the sub-field of machine learning is vast. Within the scope of this article, we consider some of the prominently utilised methods with cross-domain applications. Artificial Neural Networks (ANNs) describe the basic principles of neural networks and are commonly applied to predictive modelling problems involving the analysis and classification of non-linear relationships within datasets. Convolutional Neural Networks (CNNs) are an adaptation of ANNs specifically designed to map image data to an output class.

CNNs are commonly applied in prediction problems involving data analyses. GANs (Generative Adversarial Neural Networks) have become publicly renowned through the emergence of "deepfakes", which has yielded strong interest in deep learning methods. Opposing to the discriminative learning of ANNs and CNNs having a clear goal, generative modelling helps with understanding data and generating hypotheses. Support Vector Machines (SVM) were the standard solution to pattern recognition tasks prior to the emergence of neural networks and were used extensively in audio, video and handwriting recognition tasks.

In the next subsections, each of these will be explained briefly to allow for better understanding of security analysis of systems utilising these methods.

A. Artificial Neural Networks

FIGURE 1. EXAMPLARY ARTIFICAL NEURAL NETWORK (ANN). This network consists of three layers with a maximum width of the layers of two (corresponds to the amount of neurons in a single layer). The dots represent the neurons. The arrows from left to right indicate the data flow from the input on the left to the output on the right. The arrows from the top indicate the configuration of each neuron with weights, which were typically acquired using a training phase. The weight collection reflects the learning outcome.



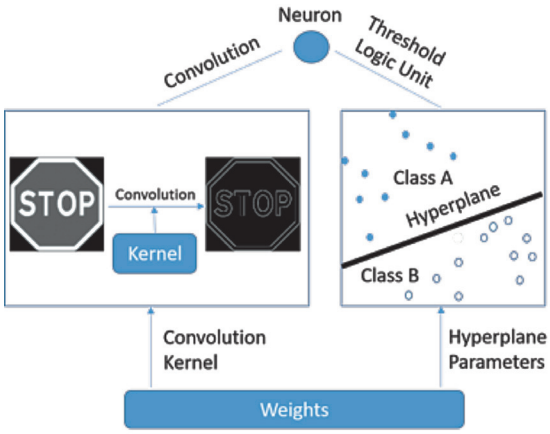
ANNs provide an abstract replication of the processes existing in the human brain. These models consist of simple atomic components called neurons, which are very limited in their individual capabilities, but which may be combined to perform more complex tasks. ANNs usually do not incorporate any task-specific rules, but instead derive the correct output from examples. Similarly to the biological model that inspired ANNs, a simple neuron may only be able to decide if an input is above a certain

threshold or not. However, collectively, a circuit of multiple neurons is capable of performing much more complex tasks. As an example, given a set of panda pictures, the ANN is able to extract a pattern of these pandas. It learns the characteristics extracted from the examples given. The system utilising the ANN will then be able to evaluate any picture with regard to these characteristics, resulting in a "match" if a sufficient number of the characteristics are met and a "mismatch" otherwise. This is called classification. Some systems are also able to provide a confidence ratio for a performed classification. However, the correctness of the classification depends greatly on the amount and variance of the training data provided. In the above example, if the panda training set only contained pandas shown from behind, the system would not be certain of the correct classification of a panda shown from the front, or may mistake an advertising pillar with a poster of black and white dots for a panda.

The peculiar strengths of ANNs are scalability and flexibility, achieved through the combination of multiple neurons. The computational capabilities are achieved through the vast connections between individual neurons. However, these multiple neurons artificially expand the "parameter space" – the space of all possible parameter combinations. Hence, the enhanced flexibility and scalability come at the price of larger training sets and higher computational power being necessary to make the neural network converge towards the correct solution.

B. Convolutional Neural Networks

FIGURE 2. CNN INVOLVING A CONVOLUTIONAL AND DENSE LAYER. The left side shows the operations of the convolutional layers, which perform the data pre-processing and feature extraction through convolution. The right side depicts the dense layers' operations that enable the CNN to classify the data based on the previously extracted features. In this example, a hyperplane is used for the classification.

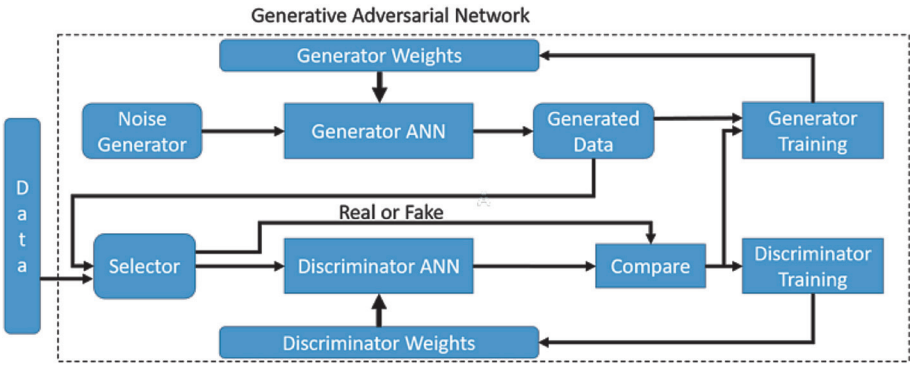


Convolutional Neural Networks (CNNs) belong to the class of "deep neural networks" (DNNs). DNNs are ANNs with multiple layers between the input and output layers. CNNs utilise two types of layers: convolutional layers and dense layers. Within the convolutional layers, each neuron processes only a small region of the input image. However, the regions are partially overlapping. This enables the network to exploit hierarchical patterns within the data and allows it to perform pre-processing and feature extractions. The dense layers are usually fully-connected ANNs used to identify patterns in the output of the convolutional layers. Dense layers are very powerful and induce a large parameter space due to the large amount of weights induced by the inter-neuron connections.

Although the convolutional layers reduce the overall parameter space, typical object detection (image classification and localisation) CNNs, such as YOLO [11], still contain over 60 million parameters. Due to the size of the parameter space, comprehensive training datasets and computational power are needed to train the network sufficiently. Therefore, pre-trained networks are available that may be used and where only the final layers must be modified to adapt to an application specific classification. This process of using pre-trained models is called "transfer learning" and is widely used.

C. Generative Adversarial Networks

FIGURE 3. VISUALISATION OF A GAN. Internally, a GAN consists of two ANNs, the generator and the discriminator, which are trained within a competitive, internal process. The generative network synthesises artificial data from random input, while the discriminator attempts to distinguish real data from the synthetic data of the generator. The selector arbitrarily selects either real or generated data and forwards this to the discriminator. The result of the discriminator is evaluated against the truth given by the selector - the evaluations outcome is utilised to train the generator and discriminator. As a result, two ANNs are trained in parallel: one produces data similar to the training data while the other is capable of identifying synthesised data.

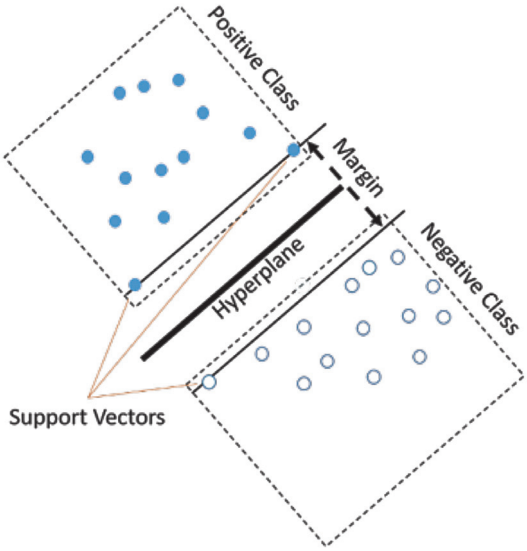


Generative Adversarial Networks (GANs) have gained much attention during the last year due to their frequent utilisation in the creation of "deepfakes". GANs consist of two competitive internal ANNs – the generator and the discriminator. These ANNs are trained in parallel in a competitive manner, which is often deployed as a zero sum or adversarial game. The discriminator tries to detect whether an input is originating from a training dataset or has been synthesised, while the generator generates adversarial samples to mislead the discriminator.

As the competitive training automatically generates feedback information, GANs do not necessarily need labelled training data. However, in order to provide reasonable output, at least the discriminator should be pre-trained on labelled data. For the creation of deepfakes, conditional GANs (cGANs) are often used, which rely on labelled data to allow a target-oriented training.

D. Support Vector Machines

FIGURE 4. VISUALISATION OF AN EXAMPLE SVM. The SVM seperates two classes of data points (blue and white) through a hyperplane while maximising the margin between the hyperplane and the nearest data points. These data points are called support vectors.



Support Vector Machines (SVMs) utilise labelled data and machine learning algorithms to perform classification and regression analysis with the help of a separating hyperplane and cluster support vectors (see Figure 4). SVMs have played a

dominant role in AI systems prior to the rise of ANNs, especially in the fields of text classification and speech recognition.

SVMs utilise mathematical concepts to define a separating hyperplane for a given set of data. Finding a separating hyperplane for a set of linearly separable clusters can be achieved through logistic regression. In order to understand non-linear relationships or solve higher-dimensional tasks, SVMs utilise "kernel tricks". The results achieved by SVMs are considered to be trustworthy and robust. However, SVMs can only perform two-class classifications (i.e. the data can only be distinguished into two categories). If more than two classes exist, algorithms must be applied that reduce the multi-class problem to several two-class problems and SVMs must be trained and executed in parallel. This limitation originates from the definition of a hyperplane, which is utilised to separate two distinct clusters. However, choosing the hyperplane to have a maximum distance between itself and the data clusters yields an inherent robustness against noise.

Some of the drawbacks of SVMs are the limitation to two-class-problems, the complexity associated with reducing multi-class problems to concurrently executable two-class-problems, the utilisation of rather complex mathematical models of kernel-functions, the necessity of labelled data input and difficulties associated with the model parameter interpretation (amongst others: finding the actual kernel function). However, SVMs are still used in various application scenarios stemming from the fields of data science, data analytics and business analytics.

3. ATTACK SURFACE

The security of AI systems and attacks directed against these systems are currently being neglected in public discussion, while the versatile utilisation of AI in varying application contexts is widely discussed. However, within the academic and technical communities, several techniques and attack vectors directed against AI systems and methods have been reported.

Currently, the most prominent attack vector categories are [12]:

- Adversarial inputs;
- Data poisoning attacks;
- Model stealing techniques.

Further attack vectors that have been identified are: model poisoning [13], model and data theft [14], data leakage [15] and neural network trojans [16]. Attack vectors

directed against the AI systems' deployment or training environment are equally applicable. These may be attack vectors directed against servers, databases, protocols or libraries utilised within the AI system. In order to allow a discussion of the vulnerabilities of AI systems, a common understanding of its attack surface must be achieved.

An attack surface allows analysts to depict the means by which an attacker may enter, extract data or manipulate the system in question. It is usually performed on software components, applications or networks in order to understand, assess and manage security risks during the design and development phase. Attack surfaces are usually designed to depict threats to a specific component or application (i.e. ignoring operators or system security issues) that stem from an outsider. However, the concept is also applicable to evaluate exposure to internal attacks [17]. Knowledge of the attack surface is invaluable in order to understand the correlations between exposure, risk and vulnerabilities [18].

A recent report of the Transatlantic Cyber Forum provided a generic, abstract attack surface claiming to cover any ML methods [19]. Opposing the attack surface derived in the aforementioned report, we will follow the OWASP guidelines on attack surface modelling, which yields an abstract yet more concrete attack surface to specific AI systems.

Currently, AI systems often lack sufficient security evaluations [20]. This may be a result of the mutually independent development of AI methods and their implementation in applications: while the application should have a security evaluation, the incorporated AI (utilised by the application through APIs or frameworks) is rarely considered in terms of its security vulnerabilities by the application developers. While the AI framework developers may follow coding standards and guidelines for secure software development, they will not evaluate the potential attack surface of an AI system utilising the framework.

As AI is expected to become ubiquitous over the next decade, the importance of understanding the vulnerabilities of AI systems and methods becomes clear. Within the following subsections, we define how attack surface modelling for AI systems should be done to include the peculiarities of these systems.

A. Data Assets

The attack surface provides information of possible entry points for an attacker as well as exit points allowing access to the systems' data. It is the result of all possible attack vectors against a system or component.

AI systems are data-driven systems that strongly depend on the data quality, authenticity and availability. Hence, data security is of particular relevance when assessing the attack surface of an AI system. Data security is usually evaluated by assessing the input validation, security at rest and security in transition. Assessing these three involves an evaluation of the impact of an attack and its likelihood of occurring. Several attack vectors directed against specific data assets in AI systems have been described (see according subsection of section 4). In addition to the AI/ML specific attack vectors, there have been reports of attacks directed against the databases holding the data assets, yielding data disclosures [21].

The impact of data alterations depends on the AI and ML methods used. Reports of minor alterations yielding majorly false classification with enormous effects in AI systems have been reported [22], while at the same time, some systems are almost ignorant to changes. Overall, the usage of sparse datasets renders the AI prone to adversarial attacks after training [23].

Furthermore, it must be recalled that for modern applications, the AI system is likely to be developed to enable concurrent processing – especially when processing large or complex data, as is the case in most AI application scenarios. A concurrent operation on data assets, however, implies the necessity for data management. The concurrent operation may either be achieved through shared databases or distributed data.

Using a database requires separate securing of that database, especially when utilising distributed and parallel computing, as the database will be addressable (through the TCP/IP stack) for external requests.

Allowing distributed data implies that the data must be kept consistent throughout the system processing entities. This is usually done by a periodic or event-triggered merging of the distributed data assets, where the data is collected from all entities. This requires authenticity of the entities involved and methods to ensure that no manipulation of the data can be performed during transportation (man-in-the-middle attacks).

B. Processing Units

Processing units within AI systems are units that are directly involved in the learning process, the data gathering or the decision making. While some attacks against the processing units will utilise data to perform the attack, other attack vectors may deploy techniques directed against the application involved (e.g. a web crawler used for data gathering is susceptible to web application vulnerabilities), the process itself or the libraries used.

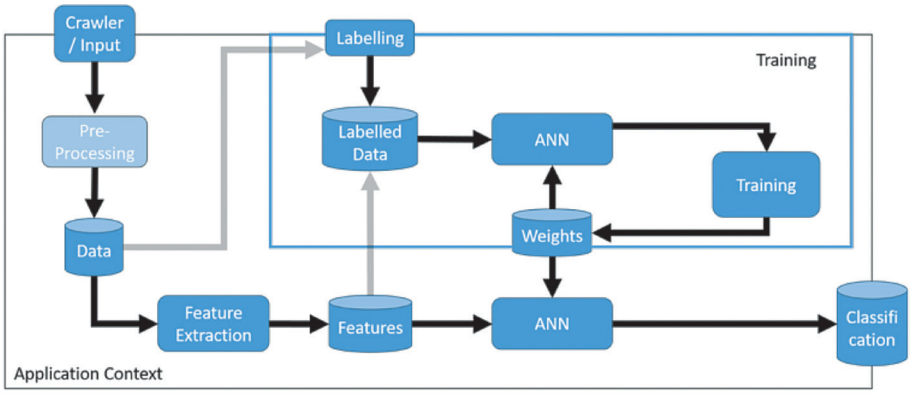
A specific type of attack combines the use of poisoned data and known vulnerabilities in the processing entities [24]. Previous attacks of this type have used audio/video files to hide malicious background operations in a steganographic manner to allow for the execution of arbitrary code. While initially considered as an attack against a specific media player, this attack utilised a meta language library vulnerability. This attack vector could have affected other applications calling the library equally, such as AI systems processing a manipulated file.

4. AI SYSTEM VULNERABILITIES

Within this section, we will use the attack surface considerations made in section 3 to define the attack surface of AI systems deploying the AI and ML methods discussed in section 2. Following the OWASP guidelines on attack surface assessments, we identify entry and exit points and briefly discuss reported and plausible attack vectors. As there are some similarities regarding the attack surfaces of ANNs, CNNs and GANs, a full explanation of an identified attack vector is given at its first encounter only. The summarising conclusion of the findings below is embedded in the overall conclusion and outlook of this paper and given in section 5.

A. ANNs

FIGURE 5. A CLASSIFICATION APPLICATION UTILISING A GENERIC ANN. The incoming data is preprocessed (reduce noise/selection of relevant material) and features are extracted. The data is labelled manually or automatically during the preprocessing. The weights of the network are adapted during the training. The final classification uses the weights derived during the training.



Looking at the overview given in Figure 5, the following attack surface points and associated vulnerabilities are identifiable:

- Crawler/Input – Entry point
  Risk of introducing unscrutinised data, data corruption and poisoning attacks. Crawlers working in a web context are web applications and susceptible to common web application vulnerabilities [25].
- Labelling – Entry point, two cases to be considered:
  Manual annotation: consideration of annotation tool vulnerabilities [26], unscrutinised data and data corruption.
  Automatic: Meta-data derived from external sources may contain malicious code, unscrutinised data, data corruption, poisoning attacks.
  Both: Attacks targeting the interface between annotation tool and ANN or targeting the functions involved in the import of the labelled data.
- Pre-processing unit – Implementation dependent, entry/exit point
  Operation on unscrutinised data, library vulnerabilities.
- Feature extraction – Implementation dependent, entry/exit point
  Operation on unscrutinised data, library vulnerabilities, database and import function vulnerabilities.
- Classifier – Exit point
  May impose threats to the overall application if data authenticity and access authorisation are not secured.
- Weights – Exit point (training); Entry point (shared weights → transfer learning)
  Authenticity of weights must be guaranteed.
  Access should be restricted to prevent theft or leakage.
  Database: database and import vulnerabilities apply.
  Volatile memory only: attack patterns against volatile memory apply.
  Shared weights: Transfer learning associated attack patterns such as NN trojans, unscrutinised data, poisoning attacks.

ANNs work with sensitive data assets. These must be protected to ensure the correctness and authenticity of the AI's output, as well as due to privacy considerations. The data assets found in AI systems utilising ANNs are:

- The data gathered itself;
- Labelled data [27] (backdoor triggers/poised data);
- Extracted features;
- Weights - Reports on volatile memory attacks exist [28], external weights obtained through model sharing may lead to trojan injections in NNs [29];
- Classification output.

Due to a lack of sufficient metrics for AI attack surfaces, it is difficult to derive a quantified and comparable assessment of the attack surface. However, it is observable that ANNs have a comparably large attack surface. The possibility of incorporating applications for the data gathering and annotation expand this attack surface even further. Overall, ANNs appear highly susceptible to a variety of cybersecurity attacks due to their complex nature of internal processing units and their frequent import/export of data requiring long-term storage.

When considering the security of the data assets, one must recall that the implementation is likely to allow concurrent processing. This implies the necessity for data management, which may either be solved through shared databases or complex merging strategies for distributed data. Both solutions imply specific attack vectors being utilisable – see section 3. A.

The application of transfer learning expands the attack surface even further, as another entry point within the ANN is established.

The above considerations provide insights into the efforts needed to secure applications utilising ANNs. The overall impression is that – without sufficient precautions being made – the attack surface of systems utilising ANNs is vast. Given the numerous reports of attack patterns directed against ANNs, this assessment appears reasonable.

B. CNNs

FIGURE 6. EXAMPLE APPLICATION UTILISING A CNN. CNNs may depict larger and more complex models as they do not have the common parameter space increase witnessed in ANNs. Pre-processing and feature extraction are performed by the CNN internally.

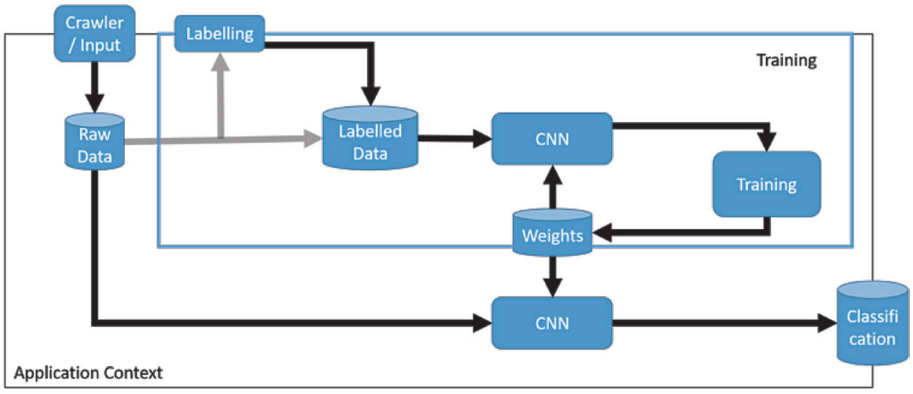


Figure 6 depicts an overview of a CNN in an abstract application context. The following attack surface points are identifiable:

- Crawler/Input – Entry point
  Susceptible to unscrutinised data, data corruption and poisoning attacks.
  Possibly susceptible to common (web) application vulnerabilities.
- Labelling – Entry point
  Manual annotation: Annotation tool vulnerabilities, unscrutinised data and data corruption.
  Automatic: Malicious meta-data, unscrutinised data, data corruption and poisoning attacks. Attacks directed against the interface between the annotation tool and the CNN (manual annotation) or against the data import of the labelled data from memory to CNN.
- Weights – Exit point (training); Entry point (shared weights, transfer learning)
  Authenticity of weights must be guaranteed.
  Access should be restricted to prevent theft or leakage.
  Database: database and import vulnerabilities apply.
  Volatile memory only: attack patterns against volatile memory apply.
  Shared weights: Due to the common utilisation of transfer learning, CNNs are particularly vulnerable to attack vectors utilising this method: Usage of externally trained weights for the CNN network may introduce logic bombs into the network [30]. This threat is hard to mitigate as it is difficult to anticipate the behaviour of CNNs based on the weights alone. The only option is to rigorously test the network with labelled data. Furthermore, NN trojans, unscrutinised data and poisoning attacks are plausible attack vectors.

CNNs work with sensitive data assets, these are:

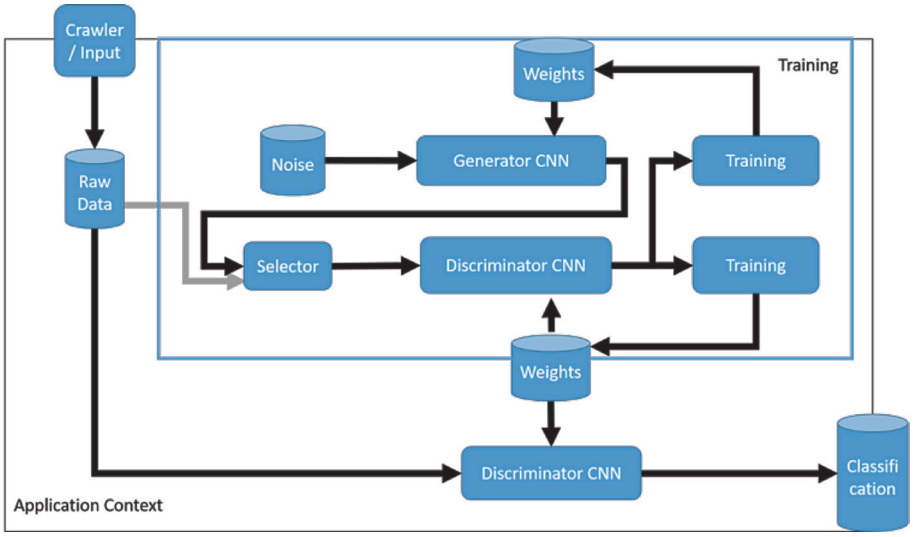
- The data gathered itself;
- Labelled data;
- Weights derived from training or through transfer learning;
- Classification results.

Within CNNs, the pre-processing and feature extraction are part of the network and not performed by separate application entities. Therefore, the data quality for CNN applications is of higher importance than for systems utilising ANNs.

In addition to the above, further attack vectors on CNNs have been reported, amongst others utilising evolutionary computing methods, evasion attacks and side-channel attacks on CNN FPGA accelerators [31].

C. GANs

FIGURE 7. AN EXEMPLARY GAN APPLICATION SYSTEM. The GAN is used to enhance the training of an already existing CNN (Discriminator CNN) for classification purposes. The Generator CNN creates additional training samples which are aimed to throw off the classification. The resulting Discriminator CNN after training is in general more robust against adversarial samples then the original one.



The attack surface is given by the systems entry/exit points, which are:

- Crawler/Input – Entry point
  See considerations in sections 4. A and B. However, for unconditional GANs such as the one shown in Figure 7, data integrity and authenticity is even more important, as no additional labels are used for the generative network. Therefore, all data points are equally important. Modification of the stochastic distribution of data may modify the behaviour of the whole GAN. The result is highly dependent on the used input data and appropriate training parameters [32].
- Weights – Exit point (training), entry point (training, shared weights, transfer learning)
  Within GANs, the weights may serve as exit and entry points.

Import/Export may be vulnerable to attacks on the interface or database used. Transfer learning is commonly used in GANs – implying GAN-based systems to be vulnerable to transfer learning attacks.

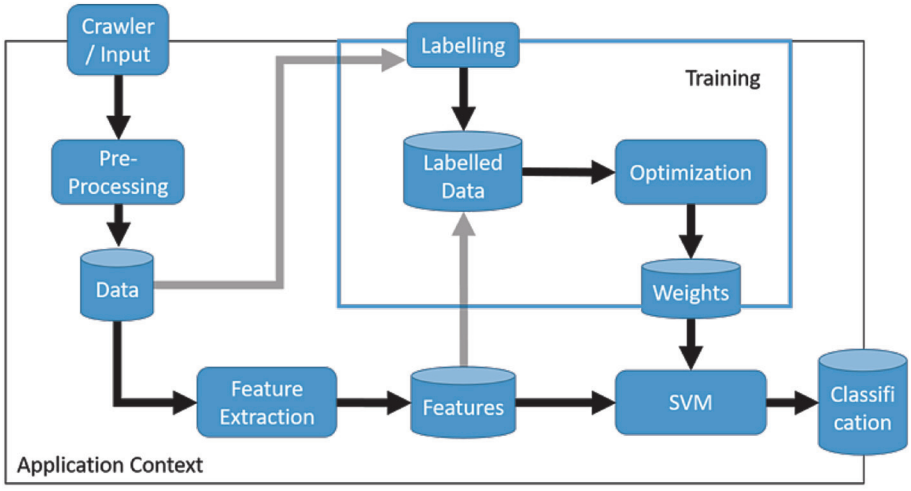
- Noise generator – (Hidden) Entry point
  The stochastic distribution of the random input used by the generator is crucial for the correct behaviour of the GAN. If the distribution is biased towards certain values, this will affect the training of both networks and may create blind spots as some data values are never generated and, therefore, the discriminator is not trained on them.
- Selector – (Hidden) Entry/exit point
  Attacks against the selector may yield a modification of the data passed. Furthermore, the selection process may be biased, yielding negative training outcomes due to an overrepresentation of real data (disabling the generator training) or an overrepresentation of artificial data (overfitting of the discriminator).
- Labelling – Entry point
  The shown unconditional GAN does not need labelled data, therefore this entry point is only present in conditional GANs. Similar to CNNs, modifying labels may negatively impact training of the discriminator. By changing labels of a specific class only, this class can be removed from the GAN altogether, preventing the generator from producing appropriate data points and also preventing the discriminator from classifying them.

Due to their composition of two CNNs operating in parallel, GANs have the same type of sensitive data assets as CNNs. Depending on the type of GAN (conditional or unconditional) labels may be present in the data (or not) and must be considered accordingly when defining the attack surface.

Most reported attacks on GANs try to reconstruct the used training data from the final model, which is called member inference attack [33]. These models can be used to generate adversarial attacks on other ML methods and also to protect them from such attacks [34].

D. SVMs

FIGURE 8. SVM APPLICATION SEPARATING LABELLED DATA INTO TWO CLASSES. Similarly to the ANNs discussed previously, pre-processing and feature extraction are performed separately from the training. The training is performed through mathematical optimisation. The SVM is executed after the training.



The following attack surface points are derivable:

- Crawler / Input – Entry point
  Unscrutinised data, data corruption and poisoning attacks.
  However, in contrast to ANNs, only a small fraction of the data defines the output. These are the support vectors identified during the training. Therefore, adversarial support vectors may heavily influence the resulting classification [35]. This type of poisoning attack is even possible in online learning environments where the SVM is continuously updated with new data [36]. Another approach uses poisoned data to prevent the training from converging through the introduction of artificially large training errors [37]. This can be used in online learning to prevent the system from updating the SVM.
- Weights – Entry point
  SVMs store a single weight per data point trained. Any data point that is not a support vector has a weight of zero. Weight modifications may therefore drastically change the output classification as it may alter the support vector identification. This allows for arbitrary output classification.

- Feature Storage – Entry/Exit point
  As the SVM is executed post-training and after the processing of the data input, it is dependent on accessing the data derived during these steps. Therefore, the feature storage is of particular importance to SVMs. An attack vector utilising this vulnerability is called the "label flip"-attack. It allows an attacker to change the label assigned to a support vector in order to change the final classification [38].
- Pre-processing and Feature Extraction – Entry points
  Data corruption and injection of malicious code in meta-data may enable an attacker to gain access to the system.

SVMs work on the following sensitive data assets:

- Raw data gathered;
- Pre-processed data;
- Features extracted;
- Labelled data;
- Weights derived – considered as the most important data points and features [39];
- Classification.

5. CONCLUSION AND OUTLOOK

Summarising the above findings and discussions, the combination of being inherently covert, their devastating impact on society and the wide unawareness of AI and ML vulnerabilities make attack vectors against these systems highly favourable for malicious cyber operators. Such attacks have already been witnessed and are being discussed in technical and academic communities but have not yet reached the public sphere, nor are application developers aware of the risk imposed by the utilisation of AI.

Despite the analyses presented in section 4, it remains difficult to provide a vulnerability hierarchy of the methods investigated regarding their susceptibility to cyber attacks. While some entry/exit points are easier to attack, others are only accessible with insider knowledge. The impact of the attack varies greatly with the data assets targeted and the specific method used. Using a preliminary approach to derive a quantifiable hierarchy based on the number of possible entry/exit points, one may observe that the number of entry/exit points is lowest in CNNs, followed by GANs and ANNs. SVMs have the same amount of identified entry/exit points as GANs. However, for AI systems, the mere number of entry/exit points is not a good

measure of the susceptibility of the technology investigated. It appears that each of the AI/ML methods investigated have specific high-value data assets, which make the system vulnerable through a combination of the data asset and a specific trait or process utilised. As an example, SVMs are highly sensitive to support vector manipulations, while GANs are exceptionally vulnerable to transfer learning attacks. The likelihood of successfully manipulating, destroying or obtaining these specific assets, traits or processes appears to give a more reliable assessment of the susceptibility than merely counting the overall number of access points. This is due to the fact that not all assets are equally important for the system to uphold its function, nor do all assets allow manipulation by an attacker or interact with the system.

In conclusion, it must be noted that AI systems are indeed susceptible to cyber attacks and that the utilisation of AI or ML methods increases any applications' vulnerability. This necessitates more sensitive use of AI and ML methods in security- or safety-sensitive applications.

Defining the attack surface of AI systems has provided information that requires further interpretation to derive the application specific risk of utilising AI/ML in the application context. Currently, only a few reports exist on attack surface metrics [40], and these are not specific to AI systems. We have seen that these systems cannot be analysed by solely investigating attack surfaces, but that the internal processing discloses particular weaknesses that are a result of the data assets used and the characteristics and processes of the methods used. Recent attacks against AI systems have shown that vulnerabilities are a result of the combination of particular AI architectures, the methods used, implementation decisions (data sharing, framework and library choices) as well as the data processing, storage and handling itself.

In order to enhance the security of AI systems, a common language to discuss the vulnerability of such systems must be installed. Furthermore, methods to reliably quantify systems' susceptibility to cyber attacks must be developed.

Policy considerations being driven by the AI community show that the need to harden AI systems against manipulations and attacks has been acknowledged within academic communities. Preliminary results from within the EU have been achieved by the Fraunhofer IAIS and the University of Bonn, who cooperated with the German Federal Office for Information Security to define a certification standard for AI, including security considerations. These results follow the EU AI HLEG and the EU AI Alliance working on the European Strategy on Artificial Intelligence.

Given the anticipated ubiquitous utilisation of AI and ML in applications over the next decade, the already existing diversity of attack vectors and the current inferiority of

countermeasures is alarming. The defence of AI systems is yet at its beginning and requires further investigation into the specific vulnerabilities of these systems [41]. Furthermore, knowledge of AI systems' vulnerabilities may become crucial to defend against cyber operations which are being carried out with the aid of AI. Such operations are currently described in modern disinformation campaigns, as well as in information and hybrid warfare with only limited countermeasures currently available. In the context of political challenges and the ongoing AI arms race, a profound knowledge of AI systems' vulnerabilities must be established to uphold cyber sovereignty.

REFERENCES

[1] Tom Simonite, "For Superpowers, Artificial Intelligence Fuels New Global Arms Race", 9 August 2017, https://www.wired.com/story/for-superpowers-artificial-intelligence-fuels-new-global-arms-race/; Catherine Clifford, "In the same way there was a nuclear arms race, there will be a race to build A.I., says tech exec", Interview with Hootsuite CEO Ryan Holmes on AI arms race, 29 September 2017, https://www.cnbc.com/2017/09/28/hootsuite-ceo-next-version-of-arms-race-will-be-a-race-to-build-ai.html.

[2] Steven Feldstein, "The Global Expansion of AI Surveillance", Carnegie Endowment for International Peace - Paper, 17 September 2019, https://carnegieendowment.org/2019/09/17/global-expansion-of-ai-surveillance-pub-79847; Bruce Schneier, "AI Has Made Video Surveillance Automated and Terrifying", Motherboard - Tech by Vice, 13 June 2019, https://www.vice.com/en_us/article/bj93z5/ai-has-made-video-surveillance-automated-and-terrifying.

[3] Jun Wu, "Artificial Intelligence and The Trader", towardsdatascience.com, 28 May 2019, https://towardsdatascience.com/artificial-intelligence-and-the-trader-500745011f53; Mike Thomas, "How AI Trading Technology is Making Stock Market Investors Smarter — and Richer - AI Trading: 17 Companies Changing The Stock Market", builtin.com, 16 March 2019, https://builtin.com/artificial-intelligence/ai-trading-stock-market-tech.

[4] Sam Daley, "Surgical robots, new medicines and better care: 32 examples of AI in healthcare", builtin.com, 23 September 2019, https://builtin.com/artificial-intelligence/artificial-intelligence-healthcare.

[5] Sandra Ponce de Leon, Cognitive World, "The Role Of Smart Grids And AI In The Race To Zero Emissions", Forbes, 20 March 2019, https://www.forbes.com/sites/cognitiveworld/2019/03/20/the-role-of-smart-grids-and-ai-in-the-race-to-zero-emissions/#b5a97221c8e3.

[6] "Computer Vision (CV) dazzle" has been inspired from dazzle camouflage used by warships in World War I and involves make-up, haircut or infrared lights to distract automated facial recognition. Further reading: Elise Thomas, "How to hack your face to dodge the rise of facial recognition tech", Wired Magazine, 1 February 2019, https://www.wired.co.uk/article/avoid-facial-recognition-software; Samantha Cole, "This Trippy T-Shirt Makes You Invisible to AI", Vice Tech, 5 November 2019, https://www.vice.com/en_us/article/evj9bm/adversarial-design-shirt-makes-you-invisible-to-ai; Jonathan Vanian, "Why Google's Artificial Intelligence Confused a Turtle for a Rifle", fortune.com, 8 November 2017, https://fortune.com/2017/11/08/google-artificial-intelligence-turtle-rifle/.

[7] Assim Rais Siddiqui, "5 Security Measures for Verified Artificial Intelligence - Find out how to ensure a secure and trusted AI system for your business", business.com, 26 August 2019, https://www.business.com/articles/security-measures-verified-artificial-intelligence/.

[8] Valecia Maclin, "Solving the challenge of securing AI and machine learning systems", Microsoft Blog, 6 December 2019, https://blogs.microsoft.com/on-the-issues/2019/12/06/ai-machine-learning-security/.

[9] Keir Giles, Kim Hartmann, Munira Mustaffa, "The Role of Deepfakes in Malign Influence Campaigns", NATO StratCom COE, ISBN 978-9934-564-50-5, September 2019, https://www.stratcomcoe.org/role-deepfakes-malign-influence-campaigns.

[10] Stephanie Kampf, Mark Kelley, "A new 'arms race': How the U.S. military is spending millions to fight fake images", CBC.ca, 18 November 2018, https://www.cbc.ca/news/technology/fighting-fake-images-military-1.4905775.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You only look once: Unified, real-time object detection", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779-788, https://doi.org/10.1109/CVPR.2016.91.

[12] Elie Bursztein, Security and Anti-Abuse Research Lead at Google, “Attacks against machine learning — an overview” Personal Site and Blog featuresing blog posts, publications and talks, May 2018, https://elie.net/blog/ai/attacks-against-machine-learning-an-overview/.
[13] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, Seraphin B. Calo, “Analyzing Federated Learning through an Adversarial Lens”, *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:634-643, 2019, http://proceedings.mlr.press/v97/bhagoji19a.html.
[14] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, Thomas Ristenpart, “Stealing Machine Learning Models via Prediction APIs”, *Proceedings of the 25th USENIX Security Symposium*, August 2016, ISBN: 978-1-931971-32-4, https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_tramer.pdf; Itay Mosafi; Eli Omid David; Nathan S. Netanyahu, “Stealing Knowledge from Protected Deep Neural Networks Using Composite Unlabeled Data”, *Proceedings of 2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, July 2019, https://ieeexplore.ieee.org/abstract/document/8851798.
[15] Taylor Larsen, “Data leakage in healthcare machine learning”, healthcare.ai, obtained 7 January 2020, https://healthcare.ai/data-leakage-in-healthcare-machine-learning/; Jason Brownlee,“Data Leakage in Machine Learning”, machinelearningmastery.com, 2 August 2016, https://machinelearningmastery.com/data-leakage-machine-learning/.
[16] Yu Ji, Zixin Liu, Xing Hu, Peiqi Wang, Youhui Zhang, “Programmable Neural Network Trojan for Pre-Trained Feature Extractor”, arXiv.com, 23 January 2019, https://arxiv.org/abs/1901.07766v1; Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang, “Trojaning Attack on Neural Networks”, Purdue University - Department of Computer Science Technical Reports, Paper 1781, 2017, https://docs.lib.purdue.edu/cstech/1781.
[17] OWASP Foundation, “Attack Surface Analysis”, OWASP Cheatsheet Series, obtained 7 January 2020, https://cheatsheetseries.owasp.org/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.html.
[18] Lily Hay Newman, “Hacker Lexicon: What Is an Attack Surface?”, wired.com, 3 December 2017, https://www.wired.com/2017/03/hacker-lexicon-attack-surface/.
[19] Sven Herping, “Securing Artificial Intelligence – Part I”, October 2019, https://www.stiftung-nv.de/sites/default/files/securing_artificial_intelligence.pdf .
[20] Dana Neustadter, “Why AI Needs Security”, Synopsys Technical Bulletin, obtained 7 January 2020, https://www.synopsys.com/designware-ip/technical-bulletin/why-ai-needs-security-dwtb-q318.html; Alexander Polyakov, “AI Security and Adversarial Machine Learning 101”, towardsdatascience.com, 23 July 2019, https://towardsdatascience.com/ai-and-ml-security-101-6af8026675ff.
[21] Jeffrey Ding, “ChinAI #47: The Sensenet Data Leak - What Actually Happened”, 25 March 2019, https://chinai.substack.com/p/chinai-47-the-sensenet-data-leak.
[22] BBC Technology, “AI image recognition fooled by single pixel change”, 3 November 2017, https://www.bbc.com/news/technology-41845878.
[23] Eykholt, Kevin, et al. “Robust physical-world attacks on deep learning visual classification, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 1625-1634, https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00175.
[24] Savia Lobo, “VLC media player affected by a major vulnerability in a 3rd library, libebml; updating to the latest version may help”, hub.packtpub.com,25 July 2019, https://hub.packtpub.com/vlc-media-player-affected-by-a-major-vulnerability-in-a-3rd-library-libebml-updating-to-the-latest-version-may-help/: CVE-2019-13615 Details, NIST National Vulnerabilities Database, 16 July 2019, https://nvd.nist.gov/vuln/detail/CVE-2019-13615.
[25] OWASP Foundation, “Web Application Security Guidance”, obtained 8 January 2020, https://www.owasp.org/index.php/Web_Application_Security_Guidance; OWASP, “OWASP Top 10 Most Critical Web Application Security Risks”, OWASP Top Ten Project, obtained 8 January 2020, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.
[26] OWASP Foundation, “Application Security Verification Standard 4.0”, March 2019, https://www.owasp.org/images/d/d4/OWASP_Application_Security_Verification_Standard_4.0-en.pdf.
[27] Duke University Press Release, “Detecting backdoor attacks on artificial neural networks”, 23 December 2019, https://ece.duke.edu/about/news/detecting-backdoor-attacks-artificial-neural-networks.
[28] Adnan Siraj Rakin, Zhezhi He, Deliang Fan, “Bit-Flip Attack: Crushing Neural Network with Progressive Bit Search”, arXiv.com, 7 April 2019, https://arxiv.org/abs/1903.12269.
[29] Zhaoyuan Yang, Naresh Iyer, Johan Reimann, Nurali Virani, “Design of intentional backdoors in sequential models”, arXiv.com, 26 February 2019, https://arxiv.org/abs/1902.09972.
[30] Gu, Tianyu, Brendan Dolan-Gavitt, and Siddharth Garg, “Badnets: Identifying vulnerabilities in the machine learning model supply chain”, *arXiv preprint arXiv:1708.06733* (2017).

[31] Jiawei Su, Danilo Vasconcellos Vargas, Kouichi Sakurai, “Attacking convolutional neural network using differential evolution”, IPSJ Transactions on Computer Vision and Applications issue 11, 22 February 2019, https://link.springer.com/article/10.1186/s41074-019-0053-3; Ya-guan Qian, Dan-feng Ma, Bin Wang, Jun Pan, Jia-min Wang, Jian-hai Chen, Wu-jie Zhou, Jing-sheng Lei, “Spot Evasion Attacks: Adversarial Examples for License Plate Recognition Systems with Convolutional Neural Networks”, arXiv.com, 28 November 2019, https://arxiv.org/abs/1911.00927; Joao Gomes, “Adversarial Attacks and Defences for Convolutional Neural Networks”, medium.com, 16 January 2018, https://medium.com/onfido-tech/adversarial-attacks-and-defences-for-convolutional-neural-networks-66915ece52e7; Lingxiao Wei, Bo Luo, Yu Li, Yannan Liu, Qiang Xu, “I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators”, in *ACSAC ‘18: Proceedings of the 34th Annual Computer Security Applications Conference*, 393–406, December 2018, https://dl.acm.org/doi/10.1145/3274694.3274696.

[32] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, “Generative Adversarial Networks: An Overview,” in IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65, Jan. 2018.

[33] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro, “LOGAN: Membership inference attacks against generative models”, in *Proceedings on Privacy Enhancing Technologies 2019*, 1, pp. 133-152; Dingfan Chen, Ning Yu,Yang Zhang, Mario Fritz, “Gan-leaks: A taxonomy of membership inference attacks against gans”, arXiv preprint arXiv:1909.03935, 2019.

[34] Samangouei, Pouya, Maya Kabkab, and Rama Chellappa. “Defense-gan: Protecting classifiers against adversarial attacks using generative models”, arXiv preprint arXiv:1805.06605, 2018.

[35] Wang, Baoyao, Peidong Zhu, Yingwen Chen, Peng Xun, and Zhenyu Zhang, “False Data Injection Attack Based on Hyperplane Migration of Support Vector Machine in Transmission Network of the Smart Grid”, Symmetry 2018, 10(5), 165, https://doi.org/10.3390/sym10050165.

[36] Xiaojun Lin and Patrick P. K. Chan, “Causative attack to Incremental Support Vector Machine”, *Proceedings of 2014 International Conference on Machine Learning and Cybernetics*, IEEE, July 2014, https://doi.org/10.1109/ICMLC.2014.7009106.

[37] Battista Biggio, Blaine Nelson, and Pavel Laskov, “Poisoning Attacks against Support Vector Machines”, in *Proceedings of the 29th International Conference on Machine Learning*, 25 March 2013, https://arxiv.org/pdf/1206.6389.pdf.

[38] Han Xiao, Huang Xiao, and Claudia Eckert, “Adversarial Label Flips Attack on Support Vector Machines”, in *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, August 2018; Ambra Demontis, Battista Biggio, Giorgio Fumera, Giorgio Giacinto, Fabio Roli, “Infinity-norm Support Vector Machines against Adversarial Label Contamination”, in *Proceedings of the 1st Italian Conference on Cybersecurity (ITASEC17)*, 2017, http://ceur-ws.org/Vol-1816/paper-11.pdf.

[39] Battista Biggio, Igino Corona, Blaine Nelson, Benjamin IP Rubinstein, Davide Maiorca, Giorgio Fumera, Giorgio Giacinto, Fabio Roli, “Security evaluation of support vector machines in adversarial environments”, Support Vector Machines Applications, pp. 105-153. Springer, Cham, 2014, https://arxiv.org/pdf/1401.7727.pdf.

[40] Pratyusa K. Manadhata, Jeannette M. Wing, “An Attack Surface Metric”, IEEE Transactions on Software Engineering (Volume: 37, Issue: 3), May-June 2011, https://doi.org/10.1109/TSE.2010.60.

[41] Adam Hadhazy, “Protecting smart machines from smart attacks”, Princeton Office of Engineering Communications, 14 October 2019, https://www.princeton.edu/news/2019/10/14/adversarial-machine-learning-artificial-intelligence-comes-new-types-attacks, quote from the text: “If machine learning is the software of the future, we’re at a very basic starting point for securing it” – Prateek Mittal, lead researcher and an associate professor in the Department of Electrical Engineering at Princeton.