Adversarial Attack's Impact on Machine Learning Model in Cyber-Physical Systems

JP Vähäkainu, MJ Lehto, AJE Kariluoto

Faculty of Information Technology
University of Jyväskylä
Jyväskylä, Finland

E-mail: Petri.vahakainu@jyu.fi; Martti.lehto@jyu.fi; anjuedka@jyu.fi

Abstract: *Deficiency of correctly implemented and robust defence leaves Internet of Things devices vulnerable to cyber threats, such as adversarial attacks. A perpetrator can utilize adversarial examples when attacking Machine Learning models used in a cloud data platform service. Adversarial examples are malicious inputs to ML-models that provide erroneous model outputs while appearing to be unmodified. This kind of attack can fool the classifier and can prevent ML-models from generalizing well and from learning high-level representation; instead, the ML-model learns superficial dataset regularity. This study focuses on investigating, detecting, and preventing adversarial attacks towards a cloud data platform in the cyber-physical context.*

Keywords: *Artificial Intelligence, Cloud Data Platform, Adversarial Attacks, Defence Mechanisms, Machine Learning*

Introduction

Artificial Intelligence (AI) is a widespread concept indicating the potential for significant societal impact. The term 'AI' has been in use for decades. AI can process a large quantity of data, and it has novel applications. AI has been utilized in various fields, for example, in construction, education, healthcare, and transportation. In the healthcare sector, AI has provided accurate diagnoses to prevent skin cancer, as well as treatment recommendations and surgical aid. In the field of Cyber-Physical Systems (CPS), AI can assist in finding anomalies and in providing predictions to reduce maintenance costs. It can also function as a defence against cyberattacks.

Accessible and correct data is paramount to have a properly functioning AI. Cybersecurity provides means to access the data in different ways. Effective cybersecurity controls provide a cyberspace infrastructure that is reliable and resilient. Lacking or absent controls lead to insecure cyberspace. Cybersecurity is applied to prevent, detect, and recover from damage to Confidentiality, Integrity, and Availability (CIA) of information in cyberspace. To use all these factors, people, processes, and technologies are used (Bayuk *et al.* 2012).

In a CPS, smart sensors automatically measure usage, functions, and variables describing the state of a building (Schmidt & Åhlund 2018). Due to decreased costs of cloud computing, IoT sensors,

and sensor networks, these techniques are becoming more common. Through provided benefits, they are gaining a foothold in a CPS context. The ample data gathered, such as water consumption, from CPS provides a significant asset to smart-service providers. The services provide added value and benefits to end-users, and they might increase cost savings and gains to organizations. Smart services can be, for example, an AI-based predictive heating or air conditioning system, or a digital caretaker. The possible advantages of the gathered data in implementing smart services are almost endless.

Cybercriminals will have a difficult time if the architecture of services is designed well, and data CIA issues have been taken seriously. Unfortunately, a significant amount of IoT devices and sensors are neither secure nor encrypted. They are not easily seen as vulnerable devices as a remarkable amount of them do not have any user interface. These rather new attack surfaces act as entry points from which cybercriminals conduct different kinds of cyberattacks. To be sure, these attackers are continuously looking for new ways to exploit vulnerabilities; and in looking for vulnerabilities to exploit, these perpetrators can find and can utilize even more sophisticated attack vectors (for example, AI-based attacks).

Several defence mechanisms presented in studies mainly focus on computer vision, and there exist only some mechanisms designed for the cybersecurity applications in mind. There is an acute need for developing defensive mechanisms to detect and to prevent cyberattacks, for example, Adversarial Attacks (AA), targeting Machine Learning (ML) model classifiers used in the CPS context. The increase of AAs towards ML-models has revealed the security and privacy issues of ML. In the smart-building context, for example, there exists a chance that, with AA, a perpetrator could fool the ML-model and gain entry to a building.

This paper is constructed in the following way. This first section, the introduction, presents the main concerns caused by adversarial attacks in the smart building's context. In the following section, the authors present the background of Artificial Intelligence and Machine Learning. Next, an explanation of cybersecurity is followed by a discussion of smart buildings and smart services. The authors then review by explaining the adversarial attack methods and their defensive methods, respectively. Finally, they discuss their findings and conclude.

Artificial Intelligence and Machine Learning

Artificial intelligence estimates a function that can include complex representations based on the data. Sometimes the results of well-made AI systems seem to mimic human-like functioning; nonetheless, the use and performance of AI depend on the quantity and quality of the data available and the desired target outcome. These systems work in a specific limited domain of input and output values and can be used to automate tasks to a certain degree. To further automate the processes of AI-systems, research fields called Machine Learning and Deep Learning (DL) have arisen. ML attempts to make AI systems more generalizable, allowing the system to learn and to adapt dynamically during the model's lifetime (Jordan & Mitchell 2015). DL is a subfield of Machine Learning. 'DL' refers to algorithms that attempt to capture the higher concepts of data.

There exist many algorithms used for AI development and learning, such as Neural Networks (NN). NN are the go-to mathematical construct to create Artificial Intelligence solutions for var-

ious tasks. This is a model that consists of interconnected nodes, which have weights associated with each connection and organized layers called the input layer, hidden layer, and output layer. Simplistically, data flows from the input layer, where it gets transformed into the model's internal representation to the hidden layer. The hidden layers operate on these values by changing the weights of the nodes with activation functions, which then direct the results to the next layers. The output layer presents the results in a more human-readable form.

Given a small sample, Generative Adversarial Neural networks (GAN) can be used to increase the amount of data to a larger one where the samples are similar the original samples. GANs are a Machine Learning system of at least two NN models put against each other. According to Radford, Metz & Chintala (2016), one of the NNs (a generator) creates outputs, and the other NN (discriminator) classifies inputs it gets from the generator and database. In an attempt to raise their performance metrics score, they will eventually improve: the classifier will label input data more accurately, and the distribution of the output of the generator will shift closer to the distribution of the original data.

Cybersecurity

The word 'cyber' comes from the Greek word κυβερεω (*kybereo*) meaning to direct, guide, and control. 'Cyber' refers to the digital world, including the surroundings and being present in our daily lives. Cybersecurity measures are associated with risk management, vulnerability patching, and improving system resilience. There are various research topics available in cybersecurity that include techniques associated with detecting network behaviour anomalies, malware, and IT security. (Lehto 2015). Cybersecurity is a collection of tools, policies, security concepts, security safeguards, guidelines, risk-management approaches, actions, training, best practices, assurance, and technologies that can be used in protecting an organization's assets (von Solms & van Niekerk 2013).

There is no globally accepted definition of 'cybersecurity', although the term is widely used. 'Cybersecurity' can be defined as a range of actions taken in defence against cyberattacks and their consequences and includes implementing the required countermeasures. Cisco (2020) defined cybersecurity as a practice of protecting systems, networks, and programs from digital attacks. Kaspersky (2020) defines cybersecurity as follows: "Cybersecurity is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks". Paloalto Networks (2020) defines cybersecurity as follows: "Cybersecurity refers to the preventative techniques used to protect the integrity of networks, programs and data from attack, damage, or unauthorised access".

Cyberattacks are often focused on accessing, changing, or destroying sensitive or critical information, blackmailing, or interrupting normal business operations. Cybersecurity is built on the threat analysis of an organization or institution. The structure and elements of an organization's cybersecurity strategy and its implementation program are based on the estimated threats and risk analyses. In many cases, it becomes necessary to prepare several targeted cybersecurity strategies and guidelines for an organization (Lehto 2015).

Preparations to counter cyber threats are essential and must focus on building sufficient protection towards the adverse effects of threats. Successful preparations can be implemented by increas-

ing common knowledge, improving operational capability, and maintaining security. Cyberattacks may not be prevented entirely. Therefore, the critical issues are being able to maintain the ability to function under attack, being able to stop the attack quickly, and being able to restore the organization's functions to its pre-incident healthy state. Finding a solution to these issues requires proper legislation and widely open discussion (Limnéll, Majewski & Salminen 2014).

Smart Buildings and Services

Smart buildings are structures that optimize energy consumption with the use of sensors and a smart meter. The meter relays information between the building and smart grid, which is a conventional grid that is utilizing IT for the optimization of energy consumption (Alam, Reaz & Ali 2012). Smart buildings with their corresponding IoT devices can be perceived as a type of CPS. According to Legatiuk and Smarsly (2018), sensors, actuators, and the building structure itself belong to the physical domain, and the functioning of the structure with proper controlling actions calculated in the cloud is known as the cyber domain of the CPS. There are different types of buildings with various kinds of usages; therefore, there are many different kinds of users. Thus, smart buildings are a CPS utilizing IoT technology to minimize energy consumption and to maximize inhabitants' satisfaction under various decision criteria.

Smart services might be best thought of as applications that automate the handling of data and somehow benefit the entire system or subsystem—be it a CPS or just a cloud-based system. The underlying technology for smart services, besides web technologies, are smart contracts. These services should consume and produce semantic data, while having elements, such as context-based adaptation and rules, which make these services smart (Maleshkova *et al.* 2019).

To rationalize the use of smart buildings, the mass of data generated by the buildings and the users ought to be used. The big data holds valuable information, such as information about the condition of a structure and utilization information, for example. The big data might be stored locally, or in the cloud; however, access to it tends to be limited, and the data is rarely used. For the designed smart services, the Data-as-a-Service platform might offer a possible solution for the data access and use. The DaaS platform operates by indexing the different data sources, such as databases, and enabling access to their data. Data does not need to be moved, which means companies and inhabitants can simultaneously offer their data for use or sale while guarding it. This would guarantee the data necessary for the development of smart services and the continued usage of the services.

Adversarial Attacks and Defence

Adversarial attacks

An adversarial attack happens when an adversarial example is sent as an input to a machine-learning model. An adversarial example can be seen as an instance to the input with features that deliberately cause a disturbance in an ML-model to deceive the ML-model into acting incorrectly and into making false predictions (Ibitoye, Shafiq & Matrawy 2019). Deep learning applications are becoming more critical each day, but they are vulnerable to AA. Szegedy *et al.* (2013) argues that making tiny changes in an image can allow someone to cheat a deep-learning model to classify the image incorrectly. The changes can be minimal and invisible to the human eye and can eventually lead to considerable differences in results between humans and trained ML-models.

The perpetrator may have perfect, limited, or zero-knowledge concerning the adversarial setting. In the case of perfect knowledge, the perpetrator has complete knowledge about features and trained models, including classifier type. When the knowledge is limited, features and the classifier are known, but not the training data of the classifier. When the knowledge is zero, the type of the classifier and the detector's model parameters are unknown. This kind of attack type is considered a black-box attack (Biggio *et al.* 2013).

Adversarial attack type can be exploratory, evasive, or causative (poisoning). Exploratory attacks are so-called white-box attacks, in which the adversary knows the classifier algorithm or training data. Exploratory attacks can also be black-box attacks, where the adversary does not know the classifier algorithm or have any knowledge about training data. In evasion attacks, the adversary focuses on specifying the data samples, which may be already misclassified by the target classifier. An example of an evasive attack is spam email generation to evade spam detection filters. The original classifier may be trained with limited training data and then retrained with an additional one. The causative attack manipulates the data at the training time and causes misclassification consequences. These attacks can be used together or individually (Sagduyu, Shi & Erpek 2019).

White-box attacks require perpetrators to understand the exact structure and parameters of the victim model in decision-time attacks or learning algorithms in poisoning attacks. In the white-box scenario, the perpetrator has full access to the victim's model, and he or she knows the ML algorithm being used, including the model's required parameters. In the poisoning attacks, he or she knows the hyperparameters of the learning algorithm. In reality, to have such detailed knowledge of the learning system seems suspicious. There may be indirect ways to obtain an adequate amount of knowledge about a learned model to apply a successful attack scenario. In case of malware evasion attack, a set of features may be public through published work. Datasets used to train the detector might be public, or there might be similar ones publicly available. The learner might use a standard learning algorithm to learn the model, such as deep NN, random forest, or Support Vector Machine (SVM) by using standard techniques to adjust hyperparameters. This may lead to the situation that the perpetrator can get a similar working detector as the actual one (Vorobeychik & Kantarcioglu 2018).

One of the emerging threats is a model extraction attack, in which a perpetrator steals a remotely deployed service provider's black-box ML-model, is given oracle prediction access, by 'reverse-engineering' it to reproduce a target model which almost perfectly replicates the victim's model (Khrishna & Papernot 2020). According to Kundu (2019), the perpetrator only requires access to predictions to conduct the attack, and the goal is to learn close approximation f' of f by extracting model parameters via querying the model with the minimal number of queries. The service provider's prediction APIs are used when performing the attack. The stolen copy of the ML-model can be used, for example, as a reconnaissance step to inform attacks in the future, to extract private information contained in the training data, to misuse paid prediction services, or to develop transferable adversarial examples to degrade the prediction quality of prediction services concerned (Khrishna & Papernot 2020). This kind of attack can pose a significant threat to CPS using ML-based predictions adjusting, for example, HVAC.

When conducting the model extraction attack, the perpetrator sends a substantial number of queries to the service provider's API. Queries are unlabeled inputs, which the perpetrator is looking

to have labelled. The API will return the victim model's prediction. The perpetrator needs to have access to the model's label, and then he or she is able to collect the predicted outputs returned by API. These queries and outputs form a pair, which can be used by the perpetrator as a training dataset to build a copy of the extracted victim model. In general, the training data is not known to the perpetrator beforehand. In model extraction, perpetrators have limited access to the victim model, and, usually, they have access to the label predicted by the model. The perpetrator is the most interested in the accuracy of the victim-extracted model (Khrishna & Papernot 2020).

ML-models are also vulnerable against Model Inversion Attacks (MIA), in which a perpetrator tries to extract private and sensitive inputs by leveraging the outputs and ML-model (Aïvodji, Gambs & Ther 2020). According to Yang *et al.* (2020), the perpetrator is expected to have either white-box or black-box access to the model. In the case of white-box access, the perpetrator knows both the internal structure of the model and the clear-text model without stored feature vectors. If the perpetrator has only black-box access, he or she can solely make prediction queries on selected feature vectors of the model and can collect the responses. The goal of the perpetrator is to extract the training data or feature vectors of the training data from the model concerned. Zheng *et al.* (2019) state that the perpetrator may conduct a model extraction attack before a model inversion attack to restore model parameters, such as model type or prediction confidence provided by service provider's API. After the model is extracted, the perpetrator can then conduct MIA to learn the training dataset, which compromises the privacy of data contributors. In addition to model inversion attack, the perpetrator can conduct an evasion attack to avoid a prediction result by modifying its query or by trying to learn a similar model utilizing query-response pairs to simulate the original model (Yang *et al.* 2020).

In the most realistic attack scenario, the perpetrator has access to the victim model's parameters only through a limited interface. Therefore, there is a need to utilize additional strategies to implement attacks without a way to access the victim model's gradients. For example, the model's gradients are not available in the case of black-box attacks. The perpetrator can train another substitute model that is different from the target model to compute the gradients needed for the attack. If the substitute and targeted models operate similarly, there is a high probability that the targeted model will misclassify the adversarial example of the substitute model.

To implement this scenario, the perpetrator needs to collect and label his or her own training set. The scenario is relatively expensive to implement due to the need for a great number of real input examples and the effort required to label each example, but the benefit is a lack of need to have access to the victim's model. If the perpetrator can send queries to the victim's model by sending it inputs and by watching the returned outputs, he or she can send inputs generated by suitable algorithms to reverse engineer a target model with small (or no) amount of training data. The perpetrator does not need to know the architecture used to create a victim's ML-model, which can be based on technologies, such as utilizing an SVM or a NN (Papernot *et al.* 2017).

In the case of black-box attacks, selecting and reducing the number of inputs the perpetrator sends to API to evade detection can be challenging. Papernot *et al.* (2017) presented a strategy (Papernot-attack) to produce synthetic inputs by using some collected real inputs. Many studies are focusing on research utilizing images as datasets (MNIST or CIFAR). In such a case, the perpetrator

can, for example, fetch several pictures of the target dataset and use the augmentation technique for each of the pictures to find new inputs that should be labelled with the API. The next step is to train a substitute by sequentially labelling and augmenting a set of training inputs. After the substitute is accurate enough, the perpetrator can launch white-box AAs, such as FGSM (Fast Gradient Sign Method) or JSMA (Jacobian Saliency Map Approach), to produce adversarial examples to be transferred to the targeted model (Goodfellow, McDaniel & Papernot 2018).

A white-box attack uses the target model's gradients in producing adversarial perturbations. FGSM was introduced by Goodfellow, McDaniel & Papernot (2018) to generate adversarial examples against NN. FGSM can be used against any ML-algorithms using gradients and weights, thus providing low computational cost. The gradient needed can be calculated by using backpropagation; and if internal weights and learning algorithm architecture is known, FGSM is efficient to execute (Co 2018). FGSM fits well for crafting many adversarial examples with major perturbations, but it is also easier to detect than L-BFGS and JSMA; therefore, L-BFGS and JSMA are stealthier perturbations, but their drawback is higher computational cost than FGSM. Defence mechanisms can prevent a relatively considerable number of FGSM and JSMA attacks, but L-BFGS is a brute-force-based white-box method, which has a high success rate despite a defence technique if time is not a critical asset (Goodfellow, McDaniel & Papernot 2018).

The perpetrator might attack against the CPS (smart building) using an AA, assuming that the defending AI system is capable of learning. For example, slowly inputting false data will eventually change the distribution of data used in the training of the new and 'improved' defensive AI. The defensive AI-model will continue to encounter like data used in training during its operation. If the AI were to protect, for example, an intelligent heating system of the CPS, it would end up functioning poorly. Depending on what is the perpetrator's end-goal, the heater could stop heating on the minimum acceptable level during times when indoor heating would be most desired in the inhabitant's view. The perpetrator might also try to fool the smart building's entry control if it relies on image data. For example, Sharif *et al.* (2016) managed to pose as another person by wearing specially designed adversarial glasses.

Adversarial attack defence mechanisms

Defending from adversarial attacks is challenging. Empirical studies state that conventional regularization strategies—such as dropout, weight decay, and distorting training data with random noise—do not present a solution to the problem. According to Samangouei, Kabkab & Chellappa (2018), several defences have been presented to reduce the effect of adversarial attacks. Defences against adversarial attacks can be divided into the following areas: 1) modifying the training data to make the classifier more robust against attacks (adversarial training augmenting the training data of the classifier with adversarial examples), 2) adjusting the classifier training process to decrease the size of gradients, and 3) reducing adversarial noise from the input samples. These methods are efficient against white-box or black-box attacks but cannot cover both types of attacks. In addition, they are designed to avert specific attack models and, therefore, are not effective against new types of attacks.

Adversarial training injects perturbed inputs, such as adversarial examples, into training data to increase the robustness of the ML-model. The goal of adversarial training is to defend from ad-

versarial perturbations by training a classifier with adversarial examples. This method can also be applied to large datasets when perturbations are crafted using fast single-step methods. Adversarial training generally attains adversarial examples by utilizing an attack, such as FGSM, and then trying to build adequate defence targeting such an attack. The trained model can indicate poor generalization capability on adversarial examples originated from other adversaries. When combining adversarial training on FGSM adversary with unsupervised or supervised domain adaptation, the robustness of the defence could be improved. Unfortunately, the robustness of adversarial training is possible to evade by applying a joint attack with indiscriminate perturbation from other models (Song *et al.* 2019).

The robustness that can be reached by adversarial training leans on the strength of the adversarial examples utilized. Training a model by using a fast non-iterative FGSM-attack produces robust protection towards non-iterative attacks, but not against PGD-based adversarial attacks. PGD-based adversarial training can be considered strong enough to sustain against powerful attacks and is considered a state-of-the-art defence model. Despite shortcomings of adversarial training, it stays among one of the few efficient methods to strengthen a network against attacks. However, its high computational complexity and cost can prevent or at least decrease utilizing it as a robust defensive method (Shafahi *et al.* 2019). According to research, a deeper understanding of adversarial training and a clear direction for further improvements are also principally missing.

Defensive distillation can be considered one of the adversarial training techniques providing flexibility to an algorithm's classification process, making the model less prone to exploitation. For example, Papernot *et. al.* (2016) have presented a defence distillation method to reduce the input variations which make adversarial crafting process more difficult, helping DNN to generalize the samples outside the training set and to reduce the effectiveness of adversarial samples on DNN. The distillation method transfers the knowledge from one architecture to another by decreasing the size of DNN. In distillation adversarial training, one model can be trained to predict the output of probabilities of another model trained on an earlier baseline standard. Defence distillation provides the advantage of being compliant with unknown threats. Usually, the most efficient adversarial defence training methods demand interminable input of signatures of known vulnerabilities and attacks into the system. The distillation provides a dynamic method requiring less human intervention. If a perpetrator has enough computing power available with the proper fine-tuning, he or she can utilize reverse engineering to find fundamental exploits. Defence distillation models are also vulnerable to poisoning attacks in which a malicious actor corrupts a preliminary training database (DeepAI).

Adversarial noise reduction from the input samples concerns the most image datasets and related applications. As time-series data, for example, weather or sensor data, is generally uni-variate or multi-variate data: the noise present in the data is in the form of missing values or different kinds of signs. In this case, missing value-related techniques, such as moving average or normalization, can be utilized. According to Moosavi-Dezfooli, Shrivastava & Tuzel (2019), measures to defend against adversarial perturbations have recently taken place by using stacked denoising auto-encoders to mitigate perturbations. The same method has been under research to denoise adversarial examples. Alternative generative models (GAN) have also been used to project malicious samples of diverse datasets. Unfortunately, these methods have mainly been applied to datasets

such as MNIST, CIFAR, or ImageNet, and there is no guarantee that attack or defence strategies could work on other kinds of data. However, an interesting observation is that an ordinary JPEG or JPEG2000 compression algorithm can act as a potential defence measure against adversarial examples, and both increase the classification accuracy of adversarial images and efficiently work in defiance of Basic Iterative Method (BIM) and FGSM (Ayademir, Temizel, & Temizel 2018).

Defence against model extraction is a challenging problem. Khrishna & Papernot (2020) presented two defence strategies to defend ML APIs against model extraction from naïve adversaries, but the models concerned are effective only in limited settings and if they are working to some degree. The first of these defences is detecting queries to be used during the model extraction attack. Another strategy is watermarking predictions made by the service provider's API to provide ownership-extracted models. According to Takemura, Yanai & Fujiwara (2020), watermarking can only verify whether an original model has been stolen through a substitute one and whether, therefore, it cannot subvert the extraction process itself. A perpetrator can keep his or her own substitute model private, making the utilization of watermarking inadequate. The perpetrator with an advanced skill can predict these defences in order to modify a suitable attack to evade the defences.

Juuti *et al.* (2019) presented a few defence strategies against model extraction. A first defence strategy is to restrict information returned to a perpetrator by modifying model prediction. Prediction probabilities can be quantized or distracted to fool the perpetrator. The first practical strategy does not provide sufficient means of defence against model extraction attacks even without using prediction probabilities. In the second strategy, the means of defence against an extraction attack is to gather requests from clients and to compute the feature space explored by the aggregated requests in order to detect the attack when the feature space exceeds a predetermined threshold. These techniques do not apply to high-dimensional input spaces nor to DNN models, as they require linearly separated prediction classes applied to decision trees. The model extraction attack can be efficiently identified by collecting stateful information of queries in ML prediction APIs. The advantage is that the defence does not need any information about the ML model or the training data, but it can be bypassed by mimicking deviations from benign distributions. Model confidentiality and stateful defence strategy can jointly protect ML models against adversarial ML attacks.

Model inversion attack is well-known in adversarial Machine Learning and is among the most malignant attacks. It contravenes data privacy providing means to infer information in the training dataset by taking advantage of confidence values disclosed with predictions from targeted models. Differential Privacy (DP) strategy has been one of the efficient countermeasures against black-box MIA, but the downside is that the strategy concerned mitigates the utility of the trained model (Evans 2018). The idea of DP is to ensure that the perpetrator cannot reckon private information from databases or from already disclosed models. DP impedes the perpetrator from perceiving the existence of specific data, for example record, by adding noise to the query responses (Bae *et al.* 2019). Even if DP can be applied to protect NNs from unintended data exposure, it is not a state-of-the-art plenary solution to stop all the threats. Homomorphic encryption combined with Bayesian NN have been developed among other options to endure MIA through secure NN inference (Yang *et al.* 2020). Reporting the rounded confidence values or predicted class labels without uncovering the confidence values could also act as a working solution.

Conclusion

The merits of this paper are the following. The authors provided discreet background information concerning Artificial Intelligence and data as a platform service in the field of cybersecurity. They also conducted a literary review on adversarial attack methods, Machine Learning, and Artificial-Intelligence-based defences. In this conclusion, the findings are discussed in the smart buildings' context and recommendations are offered to the threat posed by adversarial attacks.

Protecting smart buildings is necessary to avoid hazards caused by cyberthreats (for example, violations of privacy, data thefts, malicious acts of vandalism, insider threats, and more). Quality data from, for example, the cyber-physical system or IoT devices, is needed to train AI solutions. However, open-source IoT data for research purposes is difficult to attain. Data-as-a-Service platform and smart services might offer a solution to usability questions by helping to automate the flow of data and transactions related to it. Recently, vulnerabilities to input samples, such as adversarial examples, have been found in deep neural networks.

One way to defend dynamically is to use an ensemble of multiple differently made AI-models, which have been trained with quality data to combat either specific or generic attacks. The choice of training multiple models for the ensemble can, in some cases, improve robustness, as in the case of Ibitoye, Shafiq & Matrawy (2019). Use of ensemble and encryption methods, such as homomorphic encryption, in combination with differential privacy and neural networks, can also hinder model inversion attacks. In certain cases, model extraction attacks can be effectively recognized when stateful information is gathered from ML models or training data. Model confidentiality and stateful defence can together secure ML models against adversarial ML attacks.

The authors suggest that one might want to use defensive distillation to narrow the classification manifold. It would also be preferable to utilize GANs in the adversarial training to make the ML model learn the differences between real inputs and the adversarial attacks. Adversarial noise removal, for example, with stacked denoising autoencoders, might also work to reduce malicious input effectiveness. However, applying powerful and robust defence mechanisms on ML classifiers may cause high overhead that weakens the classifier performance. Challenges also arise when trying to transfer attacks generated for one model to other models. In the case of IoT devices, a dynamic and fit defence mechanism to detect and to prevent sophisticated adversaries should be explored. At the very least, the defence should be separated from the controlling systems.

References

Alam, MR, Reaz, MBI & Ali, MAM 2012, 'A review of smart homes—Past, present, and future', *IEEE transactions on systems, man, and cybernetics*, Part C (Applications and reviews), vol. 42, no. 6, pp.1190-203.

Aïvodji, U, Gambs, S & Ther, T 2020, *GAMIN: An adversarial approach to black-box model inversion*, Cornell University, Ithaca, NY, US, arXiv.org> cs> arXiv:1909.11835.

Ayademir, AE, Temizel, A & Temizel, TT 2018, *The Effects of JPEG and JPEG2000 compression on attacks using adversarial examples*, Cornell University, Ithaca, NY, US, arXiv.org>cs> arXiv:1803.10418

Bae, H, Jang, J, Jung, D, Ha, H & Yoon, S 2019, *Security and Privacy Issues in Deep Learning*, Cornell University, Ithaca, NY, US, arXiv.org>cs> arXiv:1807.11655v3.

Bayuk JL, Healey J, Rohmeyer P, Sachs MH, Schmidt J & Weiss J 2012, *Cyber security policy guidebook,* Wiley, Hoboken, NJ, US.

Biggio, B, Corona, I, Maiorca, D, Nelson, B, Srndic, N, Laskov, P, Giacinto, G & Roli, F 2013, *Evasion Attacks Against Machine Learning at Test Time*, Cornell University, Ithaca, NY, US, arXiv.org>cs> arXiv:1708.06131v1.

Cisco, 'What is cybersecurity?', viewed 12 July 2020, <https://www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html>.

Co, KT 2018, *Bayesian optimization for black-box evasion of Machine Learning systems*, Imperial College, London, UK.

DeepAI, 'What is defensive distillation?', viewed 10 September 2019 <https://deepai.org/machine-learning-glossary-and-terms/defensive-distillation>.

Evans, D 2018, *Adversarial Machine Learning*, University of Virginia, Charlottesville, VA, US.

Goodfellow, I, McDaniel, P & Papernot, N 2018, 'Making Machine Learning robust against adversarial inputs', *Communications of the ACM*, vol. 61, no. 7, pp. 56-66.

Ibitoye, O, Shafiq, O & Matrawy, A 2019, *Analysing adversarial attacks against deep learning for intrusion detection in IoT Networks*, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:1905.05137.

Jordan, MI & Mitchell, TM 2015, 'Machine learning: Trends, perspectives, and prospects', *Science*, vol. 349, issue 6245, pp. 255-60.

Juuti, M, Szyller, S, Marchal, S & Asokan, N 2019, *PRADA: Protecting against DNN Model stealing attacks*, Cornell University, Ithaca, NY, US, arXiv.org>cs> arXiv:1805.02628.

Kaspersky 2020, 'Cryptography definition', AO Kaspersky Lab, viewed 24 April 2020, <https://usa.kaspersky.com/resource-center/definitions/what-is-cryptography>.

Khrishna, K & Papernot, N 2020, 'How to steal modern NLP systems with gibberish?', Clevehans-blog, viewed 28 May 2020, <http://www.cleverhans.io/2020/04/06/stealing-bert.html>.

Kundu, S 2019, 'Can you trust your Machine Learning system?', National Science Foundation, viewed 28 May 2020, <https://bit.ly/3esXAES>.

Legatiuk, D & Smarsly, K 2018, 'An abstract approach towards modeling intelligent structural systems', 9th EWSHM 2018, Creative Commons CC-BY-NC licence, viewed 7 December 2020, <https://creativecommons.org/licenses/by-nc/4.0>.

Lehto, M 2015, 'Phenomena in the cyber world', *Cyber security: Analytics, technology and automation*, Springer, CH, pp. 3-29.

Limnéll, J, Majewski, K & Salminen, M 2014, 'Kyberturvallisuus, Saarijärvi' (*Cybersecurity for decision makers*), Docendo.FI, E-PUB.

Maleshkova, M, Philipp, P, Sure-Vetter, Y & Studer, R 2019, *Smart Web Services (SmartWS): The future of services on the web*, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:1902.00910.

Moosavi-Dezfooli, SM, Shrivastava, A & Tuzel, O 2019, *Divide, denoise, and defend against adversarial attacks*, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:1802.06806.

Paloalto Networks 2020, *What is cybersecurity?*, Palo Alto Networks, Inc., viewed 22 April 2020, <https://www.paloaltonetworks.com/cyberpedia/what-is-cyber-security>.

Papernot, N, McDaniel, P, Goodfellow, I, Jha, S, Celik, ZB & Swami, A 2017, 'Practical black-box attacks against deep learning systems using adversarial examples', *Proceedings of the ACM Asia Conference on Computer and Communications Security, UAE*, ACM Press, New York, NY, US.

Papernot, N, McDaniel, P, Wu, X, Jha, S. & Swami, A 2016, *Distillation as a defense to adversarial perturbations against Deep Neural Networks*, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:1511.04508.

Radford, A, Metz, L & Chintala, S 2016, *Unsupervised representation learning with Deep Convolutional Generative Adversarial Networks*, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:1511.06434v2.

Samangouei, P, Kabkab, M & Chellappa, R 2018, 'Defence-GAN: Protecting classifiers against adversarial attacks using generative models', *Proceedings of the International Conference on Learning Representations*, (ICLR 2018), Vancouver Convention Center, BC, Canada.

Sagduyu, YE, Shi, Y & Erpek, T 2019, 'IoT network security from the perspective of adversarial deep learning', Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, US.

Schmidt, M & Åhlund, C 2018, 'Smart buildings as cyber-physical systems: Data-driven predictive control strategies for energy efficiency', *Renewable and Sustainable Energy Reviews*, vol. 90, pp. 742-756, <https://doi.org/10.1016/j.rser.2018.04.013>.

Shafahi, A, Najibi, M, Ghiasi, A, Xu, Z, Dickerson, J, Studer, C, Davis, LS, Taylor, G & Goldstein, T 2019, *Adversarial training for free!*, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:1904.12843.

Sharif, M., Bhagavatula, S., Bauer, L. and Reiter, M.K., 2016, 'Accessories to a crime: Real and stealthy attacks on state-of-the-art face recognition', *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security,* pp. 1528-40.

Song, C, He, K, Wang, L & Hopcroft, JE 2019, *Improving the generalization of adversarial training with domain adaptation'*, *International Conference on Learning Representations*, New Orleans, LA, US, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:1810.00740.

Szgedy, C, Zaremba, W, Sutskever, I, Bruna, J, Erhan, D, Goodfellow, I & Fergus, R 2013, *Intriguing properties of Neural Networks*, Cornell University, Ithaca, NY, US, arXiv.org>cs> arXiv:1312.6199.

Takemura, T, Yanai, N & Fujiwara, T 2020, *Model extraction attacks against recurrent Neural Networks*, Cornell University, Ithaca, NY, US, arXiv.org>cs>arXiv:2002.00123.

von Solms, R & van Niekerk, J 2013, 'From information security to cyber security', *Computers & Security*, vol. 38, no. 13, pp. 97-102.

Vorobeychik, Y & Kantarcioglu, M 2018, *Adversarial Machine Learning,* Synthesis lectures of Artificial Intelligence and Machine Learning, vol. 12, no. 3, Morgan & Claypool Publishers, San Rafael, CA, US.

Yang, Q, Liu, Y, Cheng, Y, Kang, Y, Chen, T & Yu, H 2020, *Federated learning,* Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 13, no. 3, Morgan & Claypool Publishers, San Rafael, CA, US.

Zheng, H, Ye, Q, Hu, H, Fang, C & Shi, J 2019, 'BDPL: A boundary differentially private layer against Machine Learning model extraction attacks', *Proceedings of the 24th European Symposium on Research in Computer Security, ESORICS 2019*, Part I, pp. 66-83, doi:10.1007/978-3-030-29959-0_4.