Countering adversarial inference evasion attacks towards ML-based smart lock in cyber-physical system context

Petri Vähäkainu, Martti Lehto, Antti Kariluoto
Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
petri.vahakainu@jyu.fi
martti.lehto@jyu.fi
antti.j.e.kariluoto@jyu.fi

Abstract Machine Learning (ML) has been taking significant evolutionary steps and provided sophisticated means in developing novel and smart, up-to-date applications. However, the development has also brought new types of hazards into the daylight that can have even destructive consequences required to be addressed. Evasion attacks are among the most utilized attacks that can be generated in adversarial settings during the system operation. In assumption, ML environment is benign, but in reality, perpetrators may exploit vulnerabilities to conduct these gradient-free or gradient-based malicious adversarial inference attacks towards cyber-physical systems (CPS), such as smart buildings. Evasion attacks provide a utility for perpetrators to modify, for example, a testing dataset of a victim ML-model. In this article, we conduct a literature review concerning evasion attacks and countermeasures and discuss how these attacks can be utilized in order to deceive the, i.e., CPS smart lock system's ML-classifier to gain access to the smart building.

Keywords adversarial machine learning · defensive mechanisms · evasion attacks · cyber-physical system · ML-based CPS smart lock system

1 Introduction

A cyber-physical system (CPS), such as a smart building, utilizes technology aiming to create a safe and healthy environment for its occupants. Buildings are pertinent to smart cities, but as the number of buildings grows, it also increases security risks. Smart building technology is still in the early stages of growth and adoption increases moderately and is steadily becoming a significant business around the world. Actors and industries interested in implementing intelligent building solutions are, for example, airports, factories, hospitals, military bases, residential buildings, etc. Adoption of smart building technology brings in associated security threats consumers are usually unaware of. In the smart-building CPS context, for example, there exists a chance that, with e.g., adversarial attack, a perpetrator could fool the ML-model and gain entry to a building causing significant security threats.

A Cyber-physical system (CPS) is an interconnected network of sensors and actuators, which are controlled by a program within a cloud. The program accepts data from the sensors and calculates, based on previous events, how the actuators should be adjusted to implement changes to the flow of the system. A smart building is an example of a CPS. Cyber-physical systems tend to produce and gather an abundance of data in real-time.

Artificial intelligence (AI) is known to be an algorithm that can mimic human behavior to an extent. It is used in multiple industries and to solve many different tasks autonomously. The use requires data for training and for the necessary testing in order to validate the desired functioning of the AI. Utilizing artificial intelligence or machine learning (ML) within the cloud program of the CPS can help to improve the operation of the CPS control cycle by optimizing, for example, the energy consumption usage of a smart building. Later, we introduce a ML-assisted smart lock as access control to an imaginary smart building (CPS) functioning as an example.

The concept of cybersecurity is extensively used and remains vague and intricated. Its goal is to enable operations in cyberspace without risk of physical or digital harm (Dewar, 2014). It can be applied to various contexts, from business operations to ICT technologies. According to De Groot (2020), cybersecurity can be defined as *“the body of technologies, processes, and practices designed to protect networks, programs and data from attack, damage or unauthorized access. Cyber security may also be referred to as information technology security.”* Gartner defines cybersecurity as follows: *“Cybersecurity is the combination of people, policies, processes and technologies employed by an enterprise to protect its cyber assets”* (Gartner, 2020).

Cybersecurity is an important viewpoint for the running of the CPS, especially when concerning the machine learning algorithms. These algorithms are probabilistic, involve a vast amount of data, and their training can take a long time as well as be costly. Since ML models are also used for various tasks ranging from non-demanding choices to highly critical decisions, the protection of the models against perpetrators is indispensable.

In this article, the authors have showcased the need for artificially intelligent systems and the importance of cybersecurity against the threats involved with both AI and CPS. In chapter 2, the authors explain artificial intelligence and machine learning in more detail. In chapter 3, the authors describe adversarial machine learning, where chapter 3.1 showcases the ML testing-phase adversarial inference evasion attacks, and chapter 3.2 reviews the defenses against evasion attacks. In chapter 4, we discuss the application of evasion attacks and defenses against them in a smart building context. Lastly, chapter 5 concludes the paper.

2 Artificial Intelligence and Machine Learning

AI is a mathematical approach to estimate a function, and it can be expressed mathematically as f(x): Rn -> Rm, where f(x) is the function to be modeled, Rn represents the real input values and Rm represents the possible real output values. ML research field is needed to make AI models and systems more capable of handling new situations (Jordan, & Mitchell, 2015), because resources might have been limited during initial training, and the new situation might be from outside the original input or output domain that was used for training of the model.

Deep Learning (DL) is a subfield of ML, where the learning is done with models that have multiple layers within their structure. The additional depth can help the models to learn more complex associations within the given data than regular AI models (LeCun et al., 2015); hence DL models are called deep. AI is a very enticing choice for many different use cases, where the function to be estimated is either unknown or difficult to implement in practice, such as machine translations. In practice, the quality and quantity of data, the structure of the model, and training time, as well as the training method, affect how any AI learns to make its choices.

Neural network (NN) is a popular base model used in the development of AI solutions. The model has three layers: an input layer, hidden layer, and output layer, where data flows from the input layer through the hidden layer consisting of multiple layers, and the result is produced to the output layer. NNs are a collection of structured, interjoined nodes whose values are comprised of all the weights of the connections coming to each node. Every value of a node is inputted to an activation function, such as a rectified linear unit (ReLu). The activation function is typically the same for all the nodes in the same layer.

Long-Short Term Memory neural network (LSTM) is a special case of Recurring Neural Network (RNN) (Lipton et al., 2015), which retains output information from previous timesteps as part of the input information. The extra information can be helpful i.e. when forecasting with sequential data. Because NNs can suffer from the problems of vanishing and exploding gradients, which likely will increase with the growth of sequence size, LSTMs have three gates within each node that are used to control the information going through them (Lipton et al., 2015). These logical gates use sinh and tanh activation functions to control the flow and size of internal representations of the inputs and outputs.

3 Adversarial machine learning

In recent years, the utilization of the ML approach has been flourishing and seen an expeditious increase. ML has been used to identify relevant patterns in the information and make even more precise predictions as a function of time. ML methods have been used to, i.e., malware detection, facial and speech recognition, robotics, autonomous cars, etc. The benefits come with the disadvantages, and ML models may be vulnerable to exploitation and manipulation. This kind of risk can be considered as "AML" as systems can be fooled by perpetrators through malicious input to cause a malfunction in ML models to make incorrect assessments. During the AML attacks (Figure 1), inaccurate or misrepresentative data can be injected into an ML model at the model training phase, or malicious data may be used to swindle a trained model to make it behave abnormally and provide false predictions.

The field of AML has emerged to study vulnerabilities of the ML approach in adversarial settings and to develop techniques to make learning robust to adversarial manipulation (Vorobeychik & Kantarcioglu, 2018). AML is about learning in the presence of adversaries, and the learning can happen, i.e., in an exploratory way in testing (inference time), when the attacker aims to confuse the decision of the ML model after it has been learned (Song, 2019). Adversarial attacks have been under extensive research recently, and one prominent finding by Szegedy et al. (2013) was in the field of computer vision, revealing that a small perturbation in the form of carefully crafted input could confuse a deep neural network (DNN) to misclassify an image object. After the study, adversarial attacks have been more widely explored beyond image classification.

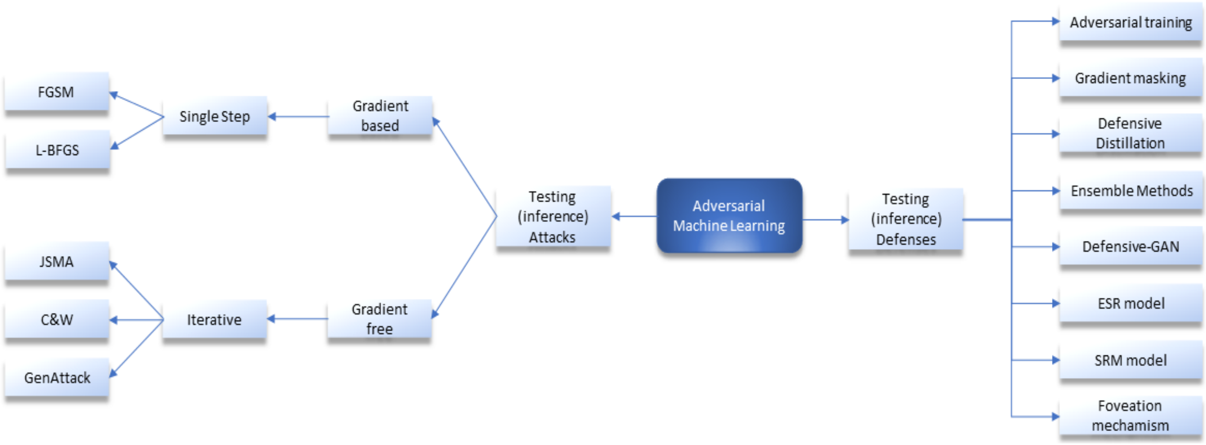

Fig. 1 Inference evasion attacks and defenses in Adversarial Machine Learning.

3.1 ML testing -phase adversarial inference evasion attacks

An evasion attack occurs when the neural network receives an adversarial example as an input, which can be considered as a deliberately perturbed appearing unmodified and precisely the same to human eyes, but is able to deceive the classifier (Moisejevs, 2019). Attacks against ML can be traced back to the year 2004 when Battista Biggio published the paper: “Wild Patterns” concerning the rise of adversarial machine learning. Research in the field of adversarial examples has been conducted to a great extent in computer vision, but they are basically applicable to any type of ML systems. For example, Sharif et al. (2016) managed to impersonate as another person by wearing particularly designed adversarial glasses. The use of glasses embodies that perturbations can occur due to other incidents than modifying images with proper adversarial techniques.

Evasion attacks can be roughly divided into gradient-based and gradient-free classes. Gradient-based attacks can be further divided into a single step and iterative attacks. Fast Gradient Sign Method (FGSM), proposed by Goodfellow et al. (2018), is a well-known single-step white-box type of an attack utilizing gradients of neural network in generating an adversarial example. In the case of adversarial images, the aim is to create an image that maximizes the loss. This can be expressed with the following equation: $adv\_x = x + \varepsilon * sign(\nabla_x J(\Theta, x, y))$ in which adv_x = adversarial example, x = original input image, y = original input label, ε = multiplier to ensure the perturbations are small, Θ = model parameters and J = loss. (TensorflowCore) FGSM attack can be used to counter any kind of ML algorithms making use of gradients and weights. The backpropagation method can be used in calculating gradients. FGSM provides low computational cost and can be effective to run if weights and learning architecture is known. FGSM applies well for crafting adversarial examples with major perturbations, but as a downside, it is easier to detect than other methods, such as L-BFGS and JSMA. (Co., 2018)

Jacobian-based saliency map algorithm (JSMA) was presented by Papernot et al. to optimize L_0 distance. JSMA attack can be used for deceiving classification models, for example, neural network classifiers, such as DNNs in image classification tasks. The algorithm is able to induce the model to misclassify the adversarial image concerned as a determined erroneous target class. (Wiyatno & Xu, 2018). JSMA is an iterative process, and in each iteration, it saturates as few pixels as possible by picking the most important pixel on the saliency map in a given image to their maximum or minimum values to fool the classifier. (Pawlak, 2020) Even though the attack alters a small number of pixels, the perturbation is more significant than L∞ attacks, such as FGSM (Ma et al., 2019). The method is reiterated until the network is cheated, or the maximal number of altered pixels is achieved. JSMA can be considered as a greedy attack algorithm for crafting adversarial examples, and it may not be useful with high dimension input images, such as images from ImageNet dataset (Ma et al., 2019).

The perturbation generated by JSMA is the gradient direction of the predicted value of the target class label, a forward derivative. The forward derivative is composed of the partial derivative value of the target class for each pixel. Liu et al. (2020) To craft an adversarial example from a given input X (example point), JSMA first computes the gradient ∇F(x) in which F denotes feedforward neural network. (Loison et al., 2020) The dimensions for the model output (number of classes) and the inputs are M and N, respectively. The Jacobian is computed by:

$$\nabla F(X) = \frac{\partial F(X)}{\partial X} = \left[\frac{\partial F_j(X)}{\partial F_j(X)}\right]_{i \in 1 \ldots N, j \in 1 \ldots M}$$

The next step is constructing a saliency map used to choose the most relevant component to perturb. The goal is to maximize the output for the target class c, $F^c(X)$, and minimize the output for the other classes $j \neq c$ (Wu et al., 2019). This can be reached by utilizing the adversarial saliency map:

$$S(X,c) = \left\{ \begin{matrix} 0, if \ \frac{\partial F_c(X)}{\partial X} < 0 \ or \ \sum_{j \neq t} \frac{\partial F_j(X)}{\partial X} > 0 \\ \frac{\partial F_c(X)}{\partial X} \left| \sum_{j \neq t} \frac{\partial F_j(X)}{\partial X} \right|, otherwise \end{matrix} \right\}$$

The adversarial attack can be created by starting from a selected example point and to iteratively perturb the example point in the direction of $S(X,c)$ by minor steps until the predicted label changes. For an untargeted attack, the prediction score is minimized for the winning class in a similar fashion. (Wu et al., 2019)

The Limited-memory-Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) is a popular non-linear box-constrained gradient-based quasi-Newtonian numerical optimization algorithm using a limited amount of memory for adversarial examples generation (Wiyatno, 2018). The algorithm can be utilized for solving high-dimensional minimization problems in where both the objective function and its gradient can be computed analytically (Coppola & Steward, 2014) Due to the costly linear search method is used to find the optimal value, especially for complex DNN networks, the algorithm is considered time-consuming and therefore impractical (Yan et al., 2017).

According to Krzaczynski, L-BFGS algorithm is for finding local extrema of functions based on Newton's method of finding stationary points of functions. The second degree of approximation can be utilized to find minimum function f(x) using the Taylor series as follows:

$$f(x_0 + \Delta x) = f(x_0) + \nabla f(x_0)^T \Delta x + \frac{1}{2} \Delta x^T \cdot H \Delta x.$$

In the formula, H denotes hessian matrix (H = B^{-1}), $f(x_0)$ is a locally modelled f at point x_0 at each algorithm iteration, $\nabla f(x_0)$ is a gradient of the function. The minimum can then be solved from the following equation:

$$\nabla f(x_0 + \Delta x) = \nabla f(x_0) = H \Delta x \ \Rightarrow \ \nabla f(x_0 + \Delta x) = 0 \ \Rightarrow \ \Delta x_0 = -B^{-1} \cdot \nabla f(x_0).$$

According to Okazaki (2014), the computational cost of the inverse Hessian matrix used in the L-BFGS algorithm is high, specifically when the objective function takes a significant number of variables. L-BFGS algorithm iteratively looks for a minimizer by utilizing approximation of the inverse Hessian matrix by information from last m iterations. The process mitigates computational time in solving large-scale problems, and in addition, it saves the memory storage. However, the L-BFGS algorithm solves the minimization problem only if the objective function F(x) and its gradient G(x) are computable.

Carlini and Wagner (C&W attack) has been presenting one of the most powerful iterative gradient-based attacks towards Deep Neural Networks (DNNs) image classifiers due to its ability to break undefended and defensively distilled DNNs on which, for example, L-BFGS and DeepFool attacks fail to find the adversarial samples. In addition, it can reach significant attack transferability. C&W attacks are optimization-based adversarial attacks, which can generate L_0, L_2 and L_∞ norm measured adversarial samples, also known by CW_0, CW_2 , and CW_∞. The attack attempts to minimize the distance between a valid and perturbed image while still causing the perturbed image to be misclassified by the model (Short et al., 2019). In many cases, it can decrease classifier accuracy near to 0 %. According to Ren et al. (2020), C&W attacks reach a 100 % success rate on naturally trained DNNs for image datasets, such as MNIST, CIFAR-10, and ImageNet. C&W algorithm is able to generate powerful adversarial examples, but computational cost is high due to the formulation of the optimization problem. The C&W attack formulates the following optimization objective:

$$\frac{min}{\delta} D(x, x+\delta) + c \cdot f(x+\delta), where\ x+\delta \in [0,1]\ and\ f(x+\delta) \leq 0.$$

In the optimization formula, δ signifies the adversarial perturbation, D means L_0, L_2 or L_∞ distance metric, and $f(x+\delta)$ denotes customized adversarial loss. The condition for function $f(x+\delta) \leq 0$ is valid only if DNN's prediction is targeted by attack. (Ren et al. 2020) This attack is to search for the smallest weighted perturbation by norms concerned in order to simultaneously force network to improperly classify the image. In the formula, c stands for a hyperparameter to balance the two parts of equation, and $f(x+\delta)$ is the loss function to measure the distance between the input image and the adversarial image. (Liu et al., 2018)

Gradient-based optimization and computation can solely be carried out in situation (white-box setting) in which a perpetrator possess full information of the model architecture and weights and in addition, full access and control over a targeted DNN. As this is not likely a real-world case, adversarial attacks in the black-box settings are considered. Gradient-based approaches usually are computationally expensive; therefore gradient-free optimization can be considered as a practical alternative. Alzantot et al. (2019) introduced GenAttack, which bases on genetic algorithms being population-based gradient-free optimization strategies. GenAttack is robust to defenses executing gradient masking or obfuscation. It is also able to craft perturbations in the black-box setting to override some gradient-altering defense methods. The algorithm can conduct successful targeted black-box attacks by querying the target model remarkably less than other comparable methods, even against large-scale high-dimensional ImageNet models, which earlier methods have had difficulties to scale to.

3.2 Defense mechanisms against evasion attacks

Several defenses have been presented in the literature to reduce the effect of adversarial attacks, such as C&W, FGSM, JSMA, and L-BFGS. The network's robustness can be improved by continuously training the model with new types of adversarial samples to make the classifier more robust against forthcoming attacks. According to Moosavi-Dezfooli et al. (2017), the number of samples to train the model doesn't solve the problem as novel types of adversarial samples emerge at all times. Luo et al. (2015) presented the 'foveation' mechanism, which can be utilized in defending against adversarial samples generated by L-BFGS and FGSM. It is assumed that by training a significant number of data sets based on the DNN classifier can be considered to be robust to image scaling and transformation changes. Confrontation mode does not have this feature.

Many potential defense mechanisms can be thought of as belonging to the group of gradient masking. These techniques generate a model without useful gradients, for example, by using the nearest neighbor classifier instead of DNN. Nearest neighbor, though, has been shown to be vulnerable to attacks based on transferring adversarial examples from smoothed nearest neighbors. Papernot et al. (2016) Gradient masking is used because most white-box attacks operate by calculating the gradient of the DNN model. (Liu et al., 2020) Therefore, if the efficient gradient cannot be calculated, the attack will not be successful. Gradient masking's primary aim is to make the gradient useless. Yanagita & Yamamura (2018) states that gradient masking is able to eliminate the valuable gradient for perpetrators, but adversarial perturbations easily transfer over most models. The models concerned can be fooled by adversarial examples crafted based on other models. A Black-box type of attack can be then utilized to overcome gradient masking defenses.

Defensive distillation can be counted to one of the adversarial training techniques providing flexibility to an algorithm's classification process, making the model less prone to exploitation. According to Carlini & Wagner (2017), it can take an arbitrary neural network, increase its robustness, and mitigate the ability to find adversarial examples from 95 % to 0.5 %. It was originally introduced to transfer learned information from one NN to another (defensive technique). Its feasibility to defend against FGSM and JSMA was demonstrated (Yanagita et al., 2019) presented a defense distillation method that can reduce the input variations making the adversarial crafting process more difficult, helping DNN to generalize the samples outside the training set and reducing 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. Defense distillation provides the advantage of being compliant with yet unknown threats. Usually, the most efficient adversarial defense 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.

As a drawback, if a perpetrator has a lot of computing power available and the proper fine-tuning, she can utilize reverse engineering to find fundamental exploits. Defense distillation models are also vulnerable to poisoning attacks in which a malicious actor corrupts a preliminary training database. (DeepAI) Defensive distillation can be evaded by the black-box approach (Papernot et al., 2016) and also with optimization attacks (Szegedy et al., 2013). Carlini & Wagner (2017) proved that defensive distillation failed against their L_0, L_2 and L_∞ attacks. These new attacks succeed in finding adversarial examples for 100 % of images on defensively distilled networks. Previously known weaker attacks can be stopped by defensive distillation, but it cannot resist more powerful attack techniques.

Empirical experiments indicate it is challenging to detect adversarial examples generated by the C&W method. However, C&W attacks can be detected by a relatively high 93 % accuracy on ImageNet-1000 by utilizing the Enhanced Spatial Rich Model (ESRM), which also provides high detection accuracy against weaker single step gradient-based FGSM attacks. The computational time of ESRM is long due to high-dimensional features. ESRM is an extended version of SPAM (Spatial Rich Model), which extracts residuals from images. Residuals can be seen as the image noise components gained by subtracting from each pixel its estimate received using a pixel predictor from the pixel's neighborhood. SPAM doesn't provide means to consider the location of modified pixels caused by adversarial attacks; therefore, estimation of the relative modification probability of each pixel is required to be done. This can be done by utilizing MPM (Modification Probability Map), which is the matrix of all pixel's modification probabilities. ESRM provides a new Markov transition probability estimation based on MPM. (Liu et al., 2018)

Defense-GAN (Generative Adversarial Networks) is a defense strategy providing sophisticated defense methods against white-box and black-box adversarial attacks used to threaten classification networks. Defense-GAN is trained to model the distribution of unperturbed images, and at inference time, it finds a close output to a given image not containing adversarial changes. Prior to sending the image to the classifier, it is projected onto the generator by minimizing the reconstruction error $||(G(z) - x||_2^2$, and the resulting reconstruction G(z) will then be passed to the classifier concerned. Training the generator to model the unperturbed training data distribution reduces potential adversarial noise. (Samangouei et al., 2018)

Defense-GAN can be utilized jointly as an add-on with any classifier without modifying the classifier structure. In addition, re-training the classifier ought not to be required, and mitigation of performance should not be prominent. The defense method concerned can be utilized to defend against any attack as it does not presume an attack model but takes advantage of the generative efficiency of GANs to reconstruct adversarial examples. Due to the GD (Gradient Descent) loop and non-linear nature of Defense-GAN, a white-box type of attacks are challenging to conduct. (Samangouei et al., 2018) The authors mentioned a conducted C&W L2-norm attack under white-box setting against a convolutional neural network classifier. Under a white-box setting, with no attack and for most of the target classifiers, accuracy was higher than 0.992, utilizing the MNIST image dataset. By conducting the attack (both FGSM and C&W) and taking advantage of the Defense-Gan defense strategy method, accuracy decreased only less than 1 %.

Defense-GAN overcomes adversarial training as a defense method, and when conducting adversarial training using FGSM in generating adversarial examples against, for example, the C&W attack, adversarial training efficiency is not sufficient. In addition, adversarial training does not generalize well against different attack methods. Increased robustness gained by using adversarial training is reached when the attack model used to generate the augmented training set is the same as that used by the perpetrator. Hence, as mentioned, adversarial training endures inefficiently against the C&W attack; therefore, a more powerful defense mechanism should be utilized. Training GANs is a remarkably challenging task, and if GANs are not trained correctly, and hyperparameters are chosen incorrectly, the performance of the defensive mechanism may significantly mitigate. (Samangouei et al., 2018)

4. ML-based smart lock system in Smart Buildings

Figure 2 showcases an example case of a smart lock system integrated into a smart building under evasion attacks. The intended use is such that in order to control access to the building, the building user is recorded on video surveillance, and each image is sent to the CPS's ML model that uses pre-taken and confirmed images from the database to compare the current user's image to the confirmed ones. A third-party service is used in the background for the comparisons, and each image before sending it to the cloud-based service is preventatively classified and perturbed. The figure also includes the API, and it is intended for users to register, sign-in, and update their own profiles, including their face images. Therefore, there are two ways a perpetrator might attack the CPS's ML model, either by means of physical adversarial evasion attacks or by using the API to send the malicious image as input to the ML model to achieve the evasion attacks.

To protect other inhabitants of the building, the ML model ought to be trained against different adversarial attacks, for example, utilizing a model that exhibits a structure of stacked ensemble. The stacked ensemble should firstly be able to take in the image as input and use defensive distillation to reduce the number of malicious pixels' effects in the image since defensive distillation is a valuable technique in order to combat unknown attacks. Then, the "new" image can be given to a group of neural networks that have different structures and have been trained with different data against multiple attacks. For example, it might be beneficial to have a CNN or LSTM that has been trained against FGSM and JSMA based adversarial attacks and to have another DNN trained with an enhanced spatial rich model to find any remaining potential C&W attack generated pixel combinations. Lastly, the second to last model of the main ML model combines the classification results and the corresponding confidence values and outputs the ruling of the other (sub)NN models. If no malicious inputs have been detected, then the input and the likeliest images from the database are sent to be compared in the cloud by the third-party vendor(s). The third-party vendor's results are used to ascertain which of the known users is in the image. This result gets inputted for the last model in the ML model, and it creates the access-allowing commands for the user to the smart building.

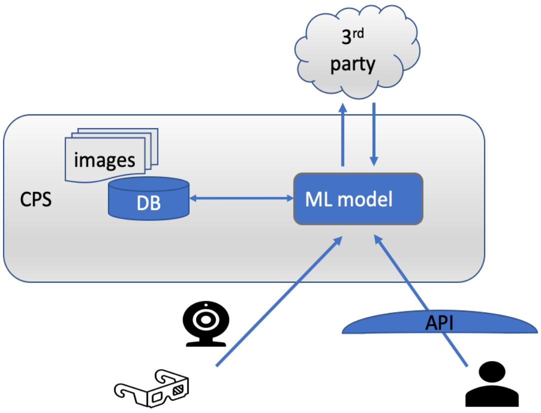

Fig. 2 Evasion attacks towards ML-based smart lock system in smart building's context.

Conclusion

In this paper, the authors reviewed the concept of adversarial machine learning and related evasion attacks. These attacks include gradient-based and gradient-free techniques, such as FGSM and C&W, respectively. The attacks can perturb the input data in such a way that the inputs seem valid for a human but mess maliciously with an artificial intelligence model.

The defenses against the adversarial evasive attacks showcased here were defensive distillation, defense-GAN, foveation, nearest neighbor clustering, Spatial Rich Model (SPAM), and enhanced spatial rich model (ESRM). To detect and prevent the adversarial attack effects, it is recommended to utilize a multitude of defensive tragedies, such as to maintain role-based access control to the models' APIs and functions. In addition, it is vital to train the desired ML models against different attacks. Unfortunately, none of the defensive methods were completely impenetrable. Thus an ensemble of AI models is recommended, even though training NNs against these attacks is time-consuming, and it costs both money and resources.

References

Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C-J., Srivastava, M., B. 2019. GenAttack: Practical Black-box Attacks with Gradient-Free Optimization. arXiv:1805.11090v3[cs.LG].

Carlini, N. & Wagner, D. 2017. Towards Evaluating the Robustness of Neural Networks. arXiv:1608.04644v2 [cs.CR].

Co., K., T. (2018). Bayesian Optimization for Black-Box Evasion of Machine Learning Systems. Imperial College London, Department of Computing.

Coppola, A. & Stewardt, B., M. 2014. LBFGS: Efficient L-BFGS and OWL-QN Optimization in R. Accessed 26.8.2020 http://cran.csiro.au/web/packages/lbfgs/vignettes/Vignette.pdf.

DeepAI. What is Defensive Distillation? Accessed 9.10.2019 https://deepai.org/machine-learning-glossary-and-terms/defensive-distillation.

De Groot, J. 2020. What is Cyber Security? Definition, Best Practices & More. Data Insider. Accessed 17.8.2020 https://digitalguardian.com/blog/what-cyber-security.

Dewar, R., S. 2014. The “Triptych of Cyber Security”: A Classification of Active Cyber Defence. 6th International Conference on Cyber Conflict, NATO CCD COE Publications, Tallinn. Accessed 17.8.2020 https://www.ccdcoe.org/uploads/2018/10/d1r1s9_dewar.pdf.

Gartner. 2020. Cybersecurity. Accessed 2.9.2020 https://www.gartner.com/en/information-technology/glossary/cybersecurity.

Goodfellow, I., McDaniel, P. & Papernot, N. (2018). Making Machine Learning Robust Against Adversarial Inputs. Communications of the ACM, 61(7), 56 – 66.

Ibitoye, O, Abou-Khamis, R., Matrawy, A. / Shafix, M., O. 2019. The Threat of Adversarial Attacks on Machine Learning in Network Security – A Survey. arXiv: 1911.02621v1 [cs.CR].

Liu, C., Ye, D., Shang, Y., Jiang, S., Li, S., Mei, Y. & Wang, L. 2020. Defend Against Adversarial Samples by Using Perceptual Hash. Computers, Materials & Continua (CMC), vol. 62, no. 3, pp. 1365-1386. DOI:10.32605/cmc.2020.07421.

Liu, J., Zhang, W. & Yu, N. 2018. CAAD 2018: Iterative Ensemble Adversarial Attack. arXiv:1811.03456[cs.CV].

Liu, J., Zhang, W., Zhang, Y., Hou, D., Liu, Y., Zha, H. & Yu, N. 2018. Detection based Defense Against Adversarial Examples from the Steganalysis Point of View. arXiv:1806.09186v2 [cs.CV].

Loison, A., Combey, T. & Hajri, H. 2020. Probabilistic Jacobian-based Saliency Maps Attacks. arXiv:2007.06032 [cs.CV].

Luo, Y., Boix, X., Roig, G., Poggio, T. & Zhao, O. 2015. Foveation-based mechanisms alleviate adversarial examples. arXiv:1511.06292.

Ma, S., Liu, Y., Tao, G., Lee, W., C., Zhang, X. 2019. NIC: Detecting Adversarial Samples with Neural Network Invariant Checking. Network and Distributed Systems Security (NDSS) Symposium 2019, San Diego, CA, USA. DOI: 10.14722/ndss.2019.23415.

Moisejevs, I. 2019. Evasion Attacks on Machine Learning (or “Adversarial Examples”). Towards data science. Accessed 30.7.2020 https://towardsdatascience.com/evasion-attacks-on-machine-learning-or-adversarial-examples-12f2283e06a1.

Moosavi-Dezfooli, S., M., Fawzi, A., Fawzi, O. & Fossard, P. 2017. Universal Adversarial Perturbations. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1765-1773.

Okazaki, N. 2014. libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). Accessed 26.8.2020 http://www.chokkan.org/software/liblbfgs.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z., B. & Swami, A. 2016. Practical Black-Box Attacks against Machine Learning. arXiv: 1602.02697[cs.CR].

Pawlak, A. 2020. Adversarial Attacks for Fooling Deep Neural Networks. Accessed 31.7.2020 https://neurosys.com/article/adversarial-attacks-for-fooling-deep-neural-networks.

Ren, K., Zheng, T., Qin, Z. & Liu, X. 2020. Adversarial Attacks and Defences in Deep Learning. Engineering, vol. 6, issue 3, pp. 346-360. DOI:10.1016/j.eng.2019.12.012.

Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M., K. (2016). Accessories to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. DOI: http://dx.doi.org/10.1145/2976749.2978392.

Samangouei, P., Kabhab, M. & Chellappa, R. 2018. Defense-GAN: Protecting Classifiers against Adversarial Attacks using Generative Models. arXiv:1805.06605v2 [cs.CV].

Short, A., Pay, T., L. & Gandhi, A. 2019. Defending Against Adversarial Examples. Sandia Report, SAND 2019-11748. Sandia National Laboratories.

Song, D. 2019. Plenary Session – Toward Trustworthy Machine Learning in Proceedings of a Workshop of Robust Machine Learning Algorithms and Systems for Detection and Mitigation of Adversarial Attacks and Anomalies, pp. 35-38.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. & Fergus, R. 2013. Intriguing Properties of Neural Networks". arXiv preprint arXiv: 1312.6199.

Teenu, S., J & Tony T. (2019). DOI: 10.4018/978-1-5225-8407-0.ch007.

TensorflowCore. 2020. Adversarial Example Using FGSM. Accessed 30.7.2020 https://www.tensorflow.org/tutorials/generative/adversarial_fgsm.

Vorobeychik, Y. & Kantarcioglu, M. 2018. Adversarial Machine Learning Synthesis Lectures on Artificial Intelligence and Machine Learning. DOI: https://doi.org/10.2200/S00861ED1V01Y201806AIM039.

Wiyatno, R. & Xu, A. 2018. Maximal Jacobian-based Saliency Map Attack. arXiv:1808.07945v1 [cs.LG].

Wu, H., Wang, C., Tyshetskiy, Y., Docherty, A., Lu, K. & Zhu, L. 2019. Adversarial Examples on Graph Data: Deep Insights into Attack and Defense. arXiv:1903.01610[cs.LG].