Defence Science Journal, Vol. 68, No. 4, July 2018, pp. 356-366, DOI : 10.14429/dsj.68.12731

A Survey of Adversarial Machine Learning in Cyber Warfare

Vasisht Duddu

Indraprastha Institute of Information Technology, Delhi - 110 020, India
E-mail: vduddu@tutamail.com

ABSTRACT

The changing nature of warfare has seen a paradigm shift from the conventional to asymmetric, contactless warfare such as information and cyber warfare. Excessive dependence on information and communication technologies, cloud infrastructures, big data analytics, data-mining and automation in decision making poses grave threats to business and economy in adversarial environments. Adversarial machine learning is a fast growing area of research which studies the design of Machine Learning algorithms that are robust in adversarial environments. This paper presents a comprehensive survey of this emerging area and the various techniques of adversary modelling. We explore the threat models for Machine Learning systems and describe the various techniques to attack and defend them. We present privacy issues in these models and describe a cyber-warfare test-bed to test the effectiveness of the various attack-defence strategies and conclude with some open problems in this area of research.

Keywords: Adversarial machine learning; Adversary modelling; Cyber attacks; Security; Privacy

1. INTRODUCTION

Machine learning (ML) and artificial intelligence are ubiquitous and have been extensively used to automate tasks and decision making processes. There has been a tremendous growth and dependence in using ML applications in national critical infrastructures and critical areas such as medicine and healthcare, computer security, spam and malware detection, autonomous driving vehicles, unmanned autonomous systems and homeland security. The critical nature of such systems and their applications demand a high level of defence against cyber attacks. While data scientists successfully automate tasks and use data mining techniques to uncover hidden, yet undiscovered knowledge from the vast unstructured data collected from disparate sources, there are serious concerns in the security issues and vulnerabilities present in data mining and ML systems. In such networks of data and knowledge sources spanning distributed databases of critical nature present in several public, private clouds, and government owned cyber infrastructures, run many ML algorithms to extract useful information and knowledge. They are highly vulnerable in the cyber ecosystem and become the weakest link in the entire chain which can compromise security of the entire system. Medical and health-care domains for instance, using ML need to ensure privacy and data leakage prevention. Recommender systems, Stock market prediction, and Sentiment analysis use ML algorithms for assessing market trends from the data and any malicious change in the data or the underlying algorithms effects the data distributions and end results. This field of ML is an important area of research owing to the growing concerns of security, privacy and over reliance of users on automated decision making. Security of ML models need to be evaluated against adversaries and defences are to be set up to ensure robust designs against adversarial attacks as shown in Fig. 1.

In this study, we explore the emerging area of adversarial machine learning (AML) which is the design of machine learning algorithms that are robust to various attacks under the constraints of the knowledge and capabilities of the adversaries. The study of AML helps in two ways: first, we can plan strategies and course of actions to model and counter against adversarial behaviour; second is to understand and model the adversary in order to strengthen our defences against the actions. These are used for red teaming in a cyberwarfare test-bed.

2. VULNERABILITIES IN MACHINE LEARNING ALGORITHMS

Vulnerabilities exist in machine learning models implemented to generate information from data. An important source of vulnerability lies in the faulty assumptions made while designing and training the ML model.

Data scientists design ML models to be robust and accurate, and they implicitly assume to preserve privacy; however this assumption is not true and leads to serious breach in privacy.

Researchers have modelled ML systems on linearly separable data and use linear function as decision function to reduce the computation complexity. This assumption increases the overall mis-classifications as an adversary can create adversarial examples to further degrade the performance of the model.

In some cases, collection of data is done in unsupervised manner and in adversarial settings like collecting data from honeypot servers. This allows attackers to carefully craft

Received : 25 November 2017, Revised : 19 March 2018
Accepted : 09 April 2018, Online published : 25 June 2018

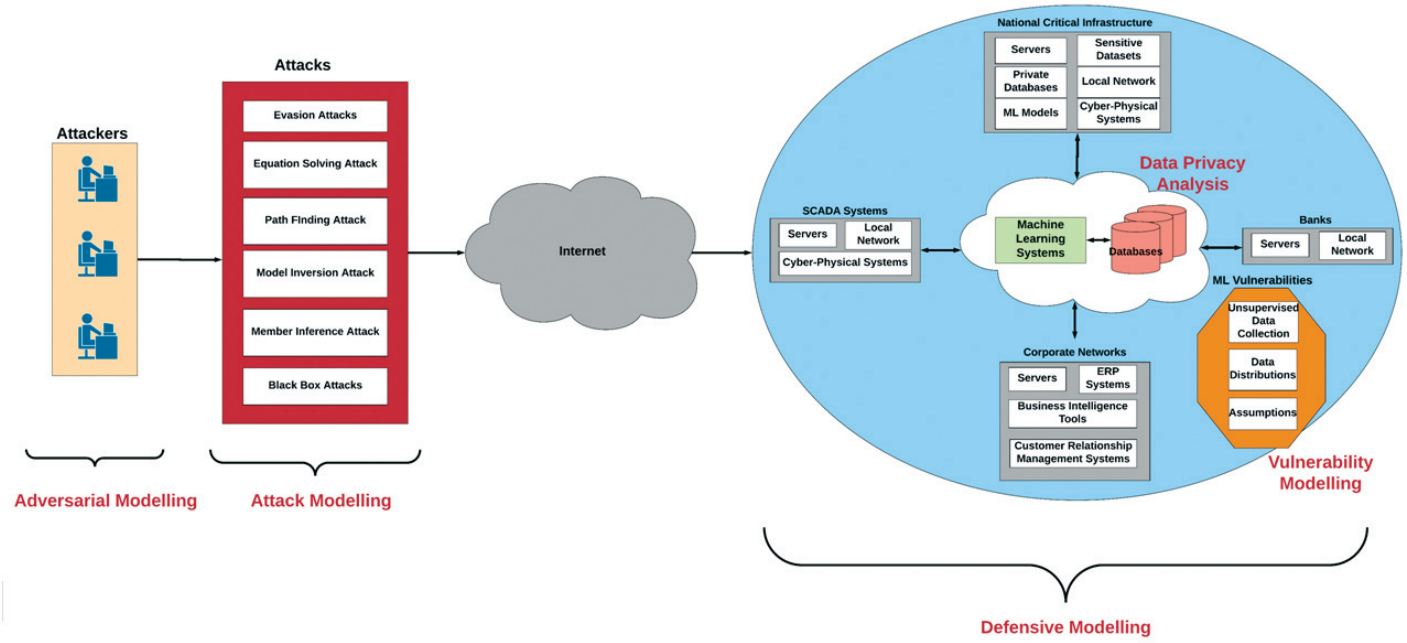

Figure 1. Major components of adversarial machine learning environment.

adversarial examples to be collected as data which may degrade the model since the adversary has direct access to the training data.

Different data instances are considered to be independent and identically distributed. Some authors, for convenience and ease of computation, assume that the features are independent of each other. However, an adversary can try to obtain the correlation between different data points and features to introduce instances from different data distribution to degrade the model's performance.

One of the major vulnerabilities in ML models is that the models perform well on testing and training data as they are usually drawn from the same underlying distribution. If the data from some other distribution is used as an input, the model will behave differently. This is the basic vulnerability that is exploited by attackers to craft adversarial examples to evade the model or degrade its performance.

3. MODELLING THE ADVERSARY

Due to the critical nature of the applications of ML, it is important to model the adversary and his strategies to attack the decision making algorithms, to represent a realistic adversary in a cyber warfare scenario.

The concept of AML was formally introduced by Huang[1], *et al.* who proposed a taxonomy of adversarial attacks and the adversary modelled using the triple: Capability, knowledge and goals.

3.1 Adversarial Capabilities

Adversarial capabilities refer to the possible impact or influence that an adversary can have by attacking the ML model. Attacks of the adversary based on the capabilities can be classified according to the following three dimensions:

- Influence
- Specificity
- Impact

Classification based on influence of adversary is based on the attempt to change the dataset or the algorithms of the target during the course of the attack. Such attacks can be further classified according to the influence as causative or exploratory.

Causative: Causative attacks alter the training process through influence over the training data. This requires the adversary to modify or influence both training and testing data.

Exploratory: Exploratory attacks do not alter the training process but use other techniques, such as probing, to discover information about training data. The adversary cannot modify or manipulate the training data and can only craft new instances based on the underlying data distribution.

The specificity of the attacks determines whether the attacks modify or effect the model as a whole based on multiple attack vectors or by using a specific attack vector to attack the model. Attacks can be classified according to specificity as:

- *Targeted*: In a targeted attack, the focus is on a single or small set of target points.
- *Indiscriminate*: An indiscriminate adversary has a more flexible goal like mis-classifying a very general class of points.

Four possible cases emerge based on the impact or effect the adversary has on the ML model[6] (Fig. 2):

- *Confidence Reduction*: Adversary tries to manipulate the training data so that the prediction confidence of the ML model reduces. This can be done when the adversary has little or no information about the model and can corrupt the decision process of the critical ML system.
- *Mis-classification*: The goal of the adversary is to mis-classify the ML model's response to an input in any way possible. This includes modifying the input to make it fall on the wrong side of the decision boundary. The attack

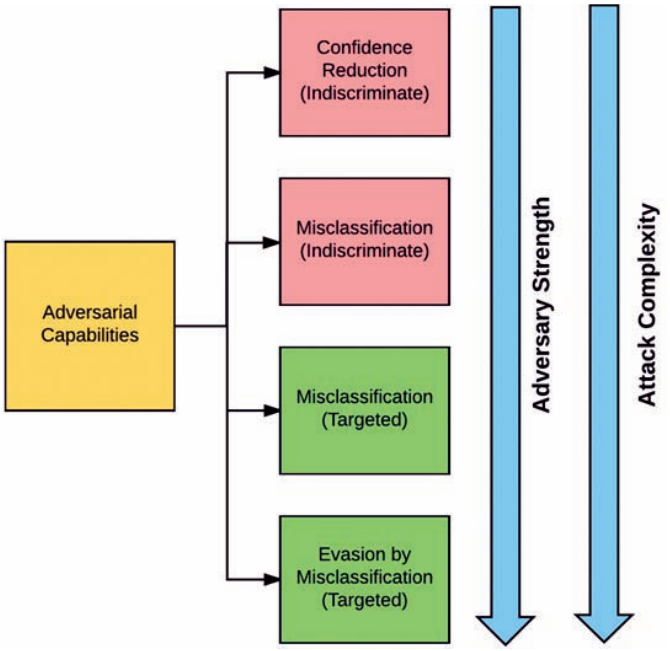

Figure 2. Impacts of adversarial capabilities.

is indiscriminate and adversary just tries to maximise the total number of mis-classifications to reduce the overall accuracy and confidence of the model.

- *Targeted Mis-classification*: The adversary generates a carefully crafted adversarial example from random noise using various algorithms and the model mis-classifies the noise as a legitimate sample. The perturbation is carefully selected unlike the previous case.
- *Source/Target Mis-classification*: An input of particular type is modified by carefully adding perturbation to be classified as a specific target class which can subvert the logic of the entire ML system. Consider an example of a ML model checking for malware and one malware instance is modified by adding perturbation to be classified as benign. This usually takes place during test time and effects only the testing data.

3.2 Adversarial Knowledge

Knowledge of the underlying ML model plays a crucial role in determining the success of the attacks by providing the adversary an opportunity to make informed decisions as shown in Fig. 3. The knowledge of the ML system can be classified into:

- Data acquisition
- Data
- Feature selection
- Algorithm and parameters
- Training and output

The adversary may have either complete or perfect knowledge of the ML system or only a partial knowledge of the system. Adversary attacks can be classified into black box attacks and white box attacks based on the knowledge about the model an adversary has.

Complete/perfect knowledge: An adversary is said to have perfect knowledge if he has access to the knowledge of data

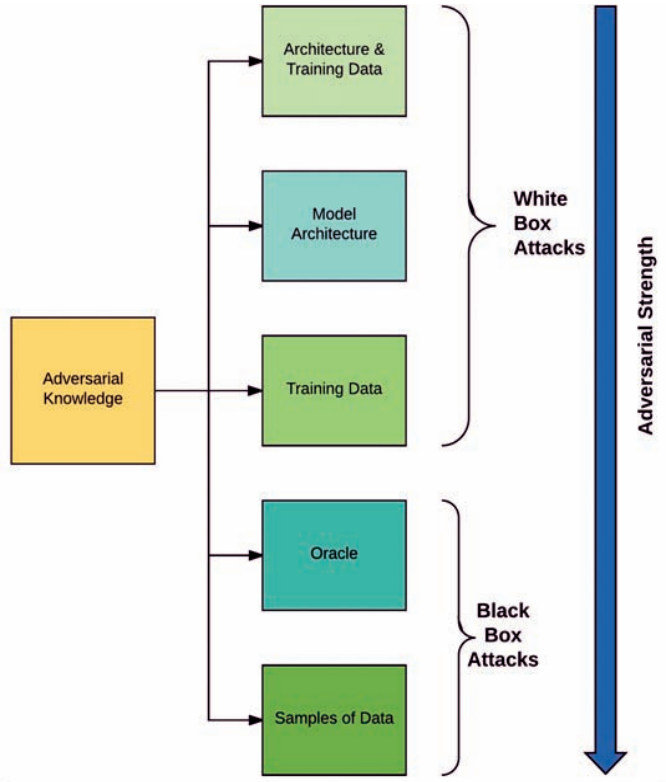

Figure 3. Adversary's knowledge.

acquisition, data, feature selection, ML algorithms and tuned parameters of the model. The attacker may or may not have access to the training data which can be easily acquired by using other knowledge. This is usually the case when the ML model is open source and everyone has access to it.

Limited Knowledge: In this case, the adversary only knows a part of the model. He does not have access to the training data and may have very limited information about the model architecture, parameters, and has access to only a small subset of the total knowledge available.

For the adversary to evolve from black box to white box, he iteratively goes through a process of learning using inference mechanisms to gain more knowledge of the model.

3.3 Adversarial Goals

Based on the goals and intent of the adversary for attacking the ML model we can classify them into the following:

Integrity violation: The adversary performs malicious activity without compromising the normal system operation but the output of the model is of attacker's choosing. Poisoning attacks are an example of integrity violation.

Availability violation: The adversary compromises the system functionality with an intent to cause a denial of service to users during the operations. One way is to maximise the mis-classification or effect the output of model significantly to degrade the performance of the model and make it to crash.

Privacy Violation: The adversary tries to gain information about sensitive user data from the ML model and also extract key information about the model architecture. Model inversion, member inference, reverse engineering and side channel on ML models are examples of such attacks.

4. ADVERSARIAL ATTACKS

In this section, we explore the various attacks on ML models. An adversary implements attacks by generating perturbed data instances called adversarial examples. A data instance may be carefully modified where the perturbations are calculated using algorithms to cause the ML classifier to mis-classify with high confidence. The goal is to construct adversarial examples x' such that it is very close to the input image x where $F(x') = T$, T being the target class, and F being the decision function. A simple indiscriminate approach is gradient ascent during training of ML model.

The fast gradient sign method (FGSM) is one of the ways to generate adversarial examples that was proposed by Goodfellow[2], *et al.* Let θ be the parameters of the model, x denotes input to the model, y denotes the targets associated with x (for a supervised learning paradigm) and $J(\theta, x, y)$ denote the cost function to train the neural network as shown in Fig. 4. The cost function can be linearised around the current value of θ, to obtain an optimal max-norm constrained perturbation of

$$\eta = \varepsilon sign \nabla_x J(\theta, x, y) \quad (1)$$

Kurakin[3], *et al*. showed that real world systems like cameras and sensors were vulnerable to adversarial examples by introducing the basic iterative method to generate adversarial images by modifying the FGSM. As an extension, they introduced label leaking effect which occurs when the accuracy on adversarial images becomes higher than the accuracy on clean images[4]. Dense adversary generation algorithm[5] generates a large family of adversarial examples to exploit semantic segmentation and object detection.

The Jacobian based saliency map approach to search for adversarial examples by modifying a small number of input pixels in an image was proposed by Papernot[6], *et al*. They compute the Jacobian of a model to identify the sensitivity of model or decision boundary. They use adversarial saliency map that contains information about the likelihood of misclassification for a given input feature.

Carlini-Wagner adversarial example[7] capable of evading all present defences including defensive distillation[72]. Given an input x, we would want to find x' and minimise $D(x, x')$ such that $F(x') = T$ and x' is valid where D is the distance function. The minimisation problem was reformulated by adding a loss function $g(x')$ that measures the closeness of $F(x')$ to T.

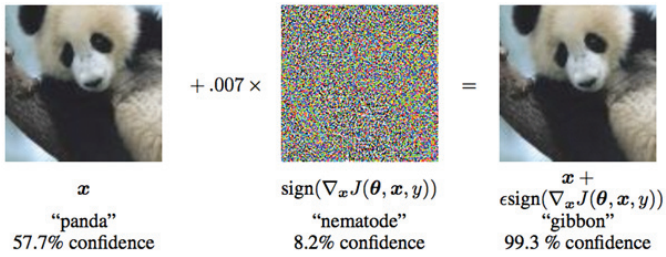

Figure 4. Generating adversarial example using fast gradient sign method[2].

4.1 Evasion Attacks

Evasion attacks evade the ML model by passing an adversarial example so that the model misclassifies. It is a test time attack that does not require accessing and manipulating the training data. The goal is to find a sample x' such that the distance from target malicious sample x_0 is minimised[81]:

$$x' = \arg\min g(x) \text{ s.t. } d(x, x_0) \leq d_{max} \quad (2)$$

Laskov[9], *et al.* study the effectiveness of evading PDF rate ML system using adversarial examples by manipulating the header fields in PDF format. An improvement was proposed[10] using an oracle which uses a function threshold to classify them as benign or malicious. A secure learning model against evasion attacks on PDF malware detection was proposed by Khorshidpour[11], *et al*. An attack on text classifiers trained using DNNs was proposed[12] using three attack strategies, namely, insertion, modification and removal of text computed using FGSM algorithm.

4.2 Poisoning Attacks

Poisoning attacks force an anomaly detection algorithm to accept an attack point that lies outside of the normal set of data instances. The attacker adds such adversarial examples to the training data so that the ML model's decision boundary can be manipulated. Poisoning is a train time attack and requires access to training data.

Kloft[13], *et al.* introduced poisoning attacks and analysed online centroid anomaly detection and adversarial noise for poisoning. In face recognition, it is possible to poison face templates with limited attacker knowledge[14]. Boiling frog attack[1], a type of poisoning attack, poisons the model over several weeks by adding small amounts of chaff. The detector is gradually acclimated to chaff and fails to identify the large amount of poisoning done incrementally. An iterative attack by selecting inputs which results in highest degradation in classification accuracy was explored using healthcare datasets[15].

4.3 Equation Solving Attack

The equation solving attack[16] is applicable on cloud providers who provide ML as a service via APIs and for models such as multi-layer perceptron, binary logistic regression and multi-class logistic regression where they are represented as equations in known and unknown variables. The goal is to use the data to find the unknown variables, which are usually the parameters used to train the models. These attacks are expected to reveal information about the model and its architecture to the attacker.

4.4 Path Finding Attack

Path-finding attacks[16] are used to traverse binary trees, multi-n-ary trees, and regression trees. In these attacks, the value of each input feature is varied till the conditions at each node are satisfied, while the input traverses the tree. The tree is traversed until a leaf is reached or an internal node with a split over a missing feature is found. The value of the leaf node is the output which reveals the path followed.

4.5 Model Inversion Attack

Fredrikson[17], *et al.* propose an algorithm that computes the optimal input feature vector close to the target feature vector using a weighted probability estimate that indicates the correct

value. The least-biased maximum a posteriori (MAP) estimate for input feature vector further minimises the adversary's incorrect predictions. This is used to create an overall model which is very close to the target model.

4.6 Black Box Attacks using Transferability Property

In black box attacks, the adversary has no access to the data and the model. The attacker can only access the oracle that returns an output for the input chosen by the attacker. ML model on cloud is an example of black box scenario where the adversary has no access to internals of the model and the training data. The service provider provides a training API using which the user can send data to the cloud to train the model and a prediction API to query the model and obtain predictions as output. In such a scenario, the adversary needs to alleviate lack of knowledge of the model and lack of knowledge of the training data.

The lack of knowledge of model can be alleviated using the property of transferability which states that samples crafted to mislead model A are likely to mislead model B. The transferability property of adversarial examples exists as they span a contiguous subspace of large dimensionality which intersect enabling transferability[18]. Transferability can be achieved in two ways[19]:

Cross-training Data Transferability: There are two different instances of data: data A and data B. The attacker trains the local model which is the same as the target model and on local model using data A while data B is used to train target model. Adversarial examples are tested on the local model which is used to attack the target model as shown in Fig. 5.

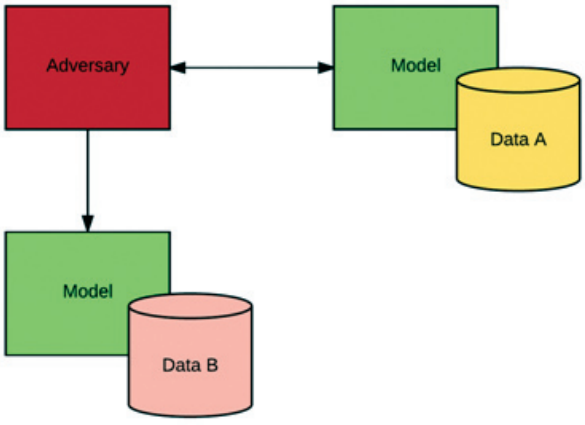

Figure 5. Cross training data transferability (Same model, different data).

Cross Technique Transferability: In this case, the attacker has access to the same data that was used to train the target model. However, he does not have access to the model internals and the local model is different from the target model. The attacker tries various model combinations to get the most optimal pair to generate the adversarial examples as shown in Fig. 6.

The lack of knowledge of data can be alleviated by using synthetic data generation. The adversary sends synthetic data to the oracle and gets a confidence score for the prediction based on which the attacker decides the validity of the synthetic data. This data and the corresponding labels given as output by the oracle are used to create a substitute for the local data.

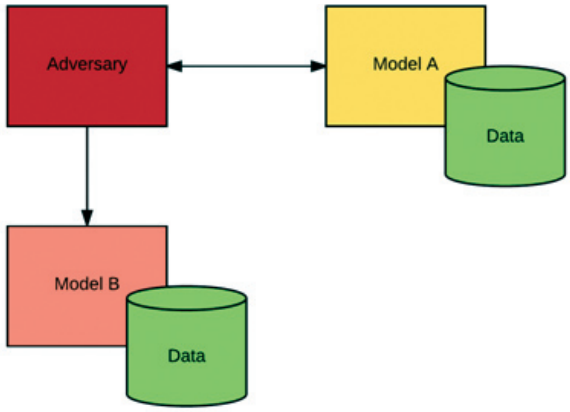

Figure 6. Cross technique transferability (Same data, different models).

Papernot[19], *et al.* use reservoir sampling to improve the previous training procedure for the substitute model. They developed a generalised algorithm for black box attacks using transferability that exploit adversarial sample transferability on broad classes of ML algorithms. This was demonstrated using a deep neural network (DNN) trained on Google and Amazon cloud services[20] as shown in Fig. 7. An ensemble based approach to generate transferable adversarial examples was proposed to attack black box models on the cloud[21]. Hayes[22], *et al.* introduce a direct attack against black-box neural networks (NNs) that uses another neural network to learn to craft adversarial examples and did not use transferability of adversarial examples unlike previous work.

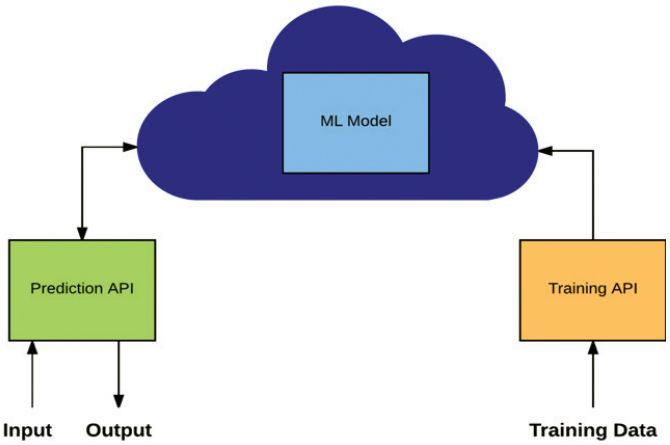

Figure 7. Cloud based black box model.

4.7 Member Inference Attack

In member inference attacks, the attacker finds if a query passed to the prediction API is part of the training set and if so leak the training data information[23].The authors implement shadow models which predict whether the input is part of the data or not.

5. ATTACKS ON VARIOUS MACHINE LEARNING PARADIGMS

Various attacks on machine learning paradigms, namely, supervised, unsupervised and reinforcement learning are discussed here.

5.1 Supervised Learning

In supervised learning, the data passed to the ML model has labels associated with each input instance. This helps in supervising the model to classify or predict values for new data instances. If the target label is a continuous range of values, it is referred to as regression problem and if the target label is a discrete value, it is referred to as classification problem. Attack models on classification models, regression models, Support vector machines (SVM) and NNs are described as follows.

Classification Models: Biggio[24], *et al*. present techniques to hide the classifier information from the adversary by introducing randomness in the decision function. Further, Biggio[25-26], *et al.* argue for improving the robustness of classifiers by an over/ under emphasis of input features of the data. The adversarial classifier reverse engineering (ACRE) learning problem[27] was introduced to learn sufficient information about a classifier so as to construct adversarial attacks by reverse engineering linear classifiers with either continuous or Boolean features.

Regression Models: Regression problems in which an adversary can exercise some control over the data generation process were first studied by Grobhans[28], *et al*. They model the problem as a Bayesian game and characterise conditions under which a unique Bayesian equilibrium point exists.

Attacks on Support Vector Machines: SVMs have been shown to be vulnerable to label flip attacks where the data labels are flipped in training data. There are two different strategies for contaminating the training set through label flipping: random and adversarial label flips[8].

Random Label Flips: The attacker randomly selects a number of samples from the training data and flip their labels.

Adversarial Label Flips: The adversary aims to find the combination of label flips which maximises the classification error on the untainted testing data. Different combinations of label flips are iterated to measure the classification error corresponding to each combination and retain that combination which gives maximum classification error and use it to attack the SVM.

A family of poisoning attacks using gradient ascent based on SVMs optimal solution have been shown to significantly increase the error[29]. A model for the analysis of label noise in support vector learning and modification of the SVM formulation that compensates for the noise by correcting the kernel matrix was suggested by Biggio[8]. A novel technique where an optimisation function was used to find label flips to maximise error classification using Tikhonov regularisation was proposed by Xiao[30]. Heuristic approach was used as an extension to improve the performance[31]. Burkard[32], *et al.* examine the targeted attack on a SVM that learns from a continuous data stream.

Attacks on Neural Networks: Szegedy[33], *et al*. were the first to identify the misclassification of NNs due to perturbed images. A maliciously trained neural network or backdoor neural network that has good performance on the user's training and validation samples, but performs poorly on specific attacker-chosen inputs was introduced[34]. The Deep Fool algorithm[35] efficiently computes perturbations that fool deep networks by minimising the distance between the adversarial example and the target example corresponding to a target class. Munoz[36], *et al.* extend the poisoning attacks to multi-class problems and propose a poisoning algorithm based on back-gradient optimisation to compute the gradient of interest through automatic differentiation to drastically reduce the attack complexity. Adversarial attacks were shown to be effective against Convolutional NN[37] and categorical and sequential Recurrent NNs using computational graph unfolding[38].

5.2 Unsupervised Learning

In unsupervised learning, data does not have any labels associated with it and only contains the input features. These are used to cluster or group the data together based on similar input features or learn a new representation of data. Attacks on unsupervised ML models can be categorised into Generative Models, Autoencoders and Clustering algorithms.

Generative Models: A generative model learns the underlying probability distributions of training data to generate and give an estimate of function fitting the distribution which enables model to generate new samples. Generative adversarial networks (GAN)[39] are a type of generative models that generate new samples by using two networks to play a game against each other. A discriminator network estimates the probability that the data is real or fake while the generative network transforms input to randomly generated samples as output and is trained to fool the discriminator network.MalGAN[40] generates adversarial malware examples, which are able to bypass black-box ML based detection models using a substitute detector. A generative network is trained to minimise the generated adversarial examples' malicious probabilities predicted by the substitute detector, making the retraining based defensive method against adversarial examples ineffective. APE-GAN[41] defends against the adversarial examples by eliminating the adversarial perturbation using a trained network and then feed the processed example to classification networks.

Autoencoders: Autoencoder is a neural network variant used for unsupervised learning where the number of neurons is same in the input and output layer. This reduces the image and represents it using less number of features (latent representation) thereby creating a sparse representation of input data for image compression, removing noisy images and creates new images. Three classes of attacks on the variational autoencoder (VAE) and VAE-GAN architectures were presented by Kos[42], *et al*. The first attack attaches a classifier to the trained encoder of the target generative model which is used to indirectly manipulate the latent representation. The second attack uses the VAE loss function to generate a target reconstruction image from the adversarial example. The third attack is based on optimising the differences in source and target latent representations.

A method to distort the input image to mislead the autoencoder in reconstructing a completely different target image was given by Tabacof[43], *et. al*. They design an attack on the internal latent representations to make the adversarial

input produce an internal representation similar to the target's representation. Makhzani[44], *et al*. propose the adversarial autoencoder (AAE), which is a probabilistic autoencoder that uses generative adversarial networks (GAN) to perform variational inference by matching the aggregated posterior of the hidden code vector of the autoencoder with an arbitrary prior distribution.

Clustering : Clustering is organising a set of data points into groups of similar features called clusters. A clustering algorithm can be formalised as a function $f : x_i$, where $i=\{1,...,n\}$, and, $C = f(D)$ is the clustering output and $D = \{x_1, x_2, ..., x_n\}$. Clustering is extensively used to infer and understand data without labels and is vulnerable to two main categories of attacks:

- *Poisoning :* Adversary aims to maximise the distance between cluster C obtained from data D and cluster C' obtained from contaminated data D' where A' is a set of adversarial samples, i.e, $C' = f(D \cup A')$.
- *Obfuscation or Bridging :* The goal is to hide attack samples in clusters without effecting the output. Bridges are formed between clusters which result in combining in clusters. Attacker's goal is to minimise the distance between C_{target} and $C' = f(D \cup A')$ [46].

These models are vulnerable mainly due to the inter-cluster distance which solely depend on the distance between closest points in the cluster which when minimised, allows attackers to form a bridge and combine the clusters[45]. The single link and complete link hierarchical clustering are vulnerable to bridging and poisoning attacks[47-48].

5.3 Reinforcement Learning

In the reinforcement learning paradigm, an agent is placed in a situation without knowledge of any goals or other information about the environment. For every action made by the agent, it receives a feedback from the environment in the form of a reward. The agent tries to maximise the reward by optimising its actions over time and the agent learns to achieve its goals. In an adversarial setting, there are multiple agents and an agent wins a game when it is given a positive reinforcement and its opponent is given negative reinforcement. Maximising reward corresponds directly to winning games and over time the agent learns to act so that it wins the game.

Uther[49], *et al.* introduce algorithms to handle the multi-agent, adversarial, and continuous-valued aspects of the domain by extending prioritised sweeping that allows generalisation of learnt knowledge over neighbouring states in the domain and to allow the handling of continuous state spaces. Behzadan[50], *et al.* establish that reinforcement learning techniques based on Deep Q-Networks (DQNs) are vulnerable to adversarial input perturbations and verify using the transferability of adversarial examples across different DQN models. They present attacks that enable policy manipulation and induction in the learning process of DQNs. Huang[51], *et al.* show that adversarial attacks are also effective when targeting neural network policies in reinforcement learning using transferability across policies to attack the Reinforcement Learning model. A method for reducing the number of adversarial examples that need to be injected for a successful attack based on the value function was proposed by Kos[52]. It was observed that retraining on random noise and FGSM perturbations improves the resilience against adversarial examples.

Lin[53], *et al.* introduce two tactics, strategically timed attack and the enchanting attack, to attack reinforcement learning agents using adversarial examples. In the strategically-timed attack, the adversary aims at minimising the agent's reward by attacking the agent at a small subset of time steps. In the enchanting attack, the adversary aims at luring the agent to a designated target state by combining a generative model to predict the future states and a planning algorithm to generate a preferred sequence of actions for luring the agent.

6. PRIVACY PRESERVING MACHINE LEARNING

Privacy preserving techniques enable to use ML on data without knowing underlying content of user's data. We study various privacy preserving models that have been proposed to ensure the protection of sensitive data. One of the main reasons for leakage of information through ML models is due to over fitting due to which generalisation becomes very important. Privacy preserving ML has followed three major directions:

- Randomisation algorithms
- Secure multi-party computation
- Homomorphic encryption (HE)

In CryptoNets[54-55], the authors perform neural network computations on data encrypted using HE and used approximations to evaluate the Sigmoid, ReLU and max pooling. The computation is slow due to the noise generated from HE and security parameters of HE should be considered carefully based on the noise. Rouhani[56], *et al.* propose a method to perform DL computation using garbled circuits (GC) and adopt pre-processing techniques to reduce the GC runtime by mapping the NN to a lower dimension. Ohrimenko[57], *et al*. propose a solution for secure multiparty ML by using trusted Intel SGX processors and used oblivious protocols between client and server where the input and outputs are blinded. Mohassel[58], *et al.* use a two server model and distribute the data into two parts for each server. The authors developed new privacy preserving protocols for linear regression, logistic regression and NNs and used garbled circuits for privacy and arithmetic with pre-computed triplets.

Differential privacy has been explored to ensure privacy guarantees for ML models for non-convex objective functions using differentially private stochastic gradient descent[59,61].

Shokri[60], *et al*. designed a model where participants use parameter sharing, allowing participants to benefit from other participants' models without explicit sharing of training inputs. After each round of training, participants asynchronously share with each other the gradients they computed for some of the parameters.

7. DEFENCES AGAINST ADVERSARIAL ATTACKS

Many approaches to building defences against adversarial attacks have been proposed over the past few years. We present different possible defences that have been proposed over the

years and discuss their shortcomings.

Gradient masking is based on the idea that if the model is non-differentiable or the model's gradient is zero at data points, then gradient based attacks are ineffective. Two major types of gradient masking are Gradient hiding and Gradient smoothing. Gradient Hiding uses models that are non-differentiable and are highly non-linear which prevent the adversary from finding the derivative. Gradient smoothing reduces the effectiveness of white-box attacks by smoothing out the model's gradient, leading to numerical instabilities in attacks such as the FGSM. However, in both white-box and black-box settings, models are still vulnerable even after using gradient masking[62].

Papernot[63-64], *et al*. designed a defence based on distillation technique where the authors leverage the softmax layer of neural network.

$$F(x) = \frac{e^{\frac{z_i(X)}{T}}}{\sum_j e^{\frac{z_j(X)}{T}}} \quad (3)$$

A low value of temperature parameter *T* will result in high confidence but discrete probabilities while a high value of *T* will reduce confidence of prediction but smooth out the probability distribution which makes crafting of adversarial examples hard. Carlini[65], *et al*. argued that the softmax layer and function used does not change output even if input is changed beyond certain values which was not considered in defensive distillation. They suggested dividing the inputs to the softmax by *T* before passing them to the function. To make it more robust, Papernot[66], *et al*. improved the defence to extended defensive distillation and modified their previous defensive distillation to address the numerical instabilities in the previous model and attacks like black box attacks using transferability. They modified the algorithm and instead of using the probabilities from first model they measured the uncertainties in classifying output using dropout inference.

Szegedy[33], *et al.* increase the model's robustness by injecting adversarial examples to the training data referred to as adversarial training which was extended to ensemble adversarial training that additionally augments training data with perturbed inputs transferred from a number of fixed pre-trained models[67]. Adversarial training to the text domain was explored by applying perturbations to the word embedded in a recurrent neural network[68].

Xu[69], *et al*. detect adversarial examples by reducing the colour depth of each pixel in an image, and spatial smoothing to reduce the difference among individual pixels. They compare the model's output with and without using feature squeezing and differentiate between adversarial or benign based on the output. A safety net architecture was proposed by Lu[70], *et al*. that consists of the original classifier and an adversary detector which looks at the internal state of the later layers in the original classifier to detect adversarial examples. Similar work was explored Metzen[71]. Reject on negative impact (RONI) defence[72] is a technique that measures the empirical effect of each training instance and eliminates from training those points that have a substantial negative impact on classification accuracy.

Data transformations like dimensionality reduction using Principal component analysis and data anti-whitening to enhance the resilience of ML models were explored by Bhagoji[73], *et al*. However, adversarial examples can be made robust to data transformations like rescaling, translation, and rotation and an approach that produces images that remain adversarial examples even after transformations[74].

The security of linear classifier itself can be improved by using evenly weighted feature weights as this would require the attacker to manipulate more features to evade detection[75].

Feature selection methods are also compromised under attack[76]. An adversary-aware feature selection model that can improve classifier security against evasion attacks was proposed by selecting a feature subset that maximises the generalisation capability of the classifier[77]. It includes forward selection and backward elimination wrapping algorithms, which iteratively add or delete a feature from the current candidate set. Feature squeezing techniques successfully detect the recent Carlini-Wagner adversarial examples[69].

The various defences described in this section are specific to models using a particular learning algorithm. As a result, a defence mechanism that is applicable to one model is not applicable to some other model. However, there is no silver bullet to defend all ML systems against adversarial attacks.

8. DISCUSSIONS AND CONCLUSIONS

Cyberwargames are designed to examine how organisations and critical response teams respond to realistic/ simulated cyber crises and highly skilled adversaries. The wargaming process comprises of the identification, defence, response, and recovery phases to a cyberattack in depth. Cyberwargames that use Game Theory to model the attackers and defenders are designed by setting up a cyber test-bed to exercise cyberattack scenarios on a network environment[78-80]. In the game theoretic framework, two approaches have been used: a probabilistic framework and a Bayesian belief framework where the attack and defender try to anticipate the opponent's strategy with complete and incomplete information with learning. In this paper, we describe the various components of cyber attacks in adversarial machine learning environments namely: Vulnerabilities of ML models in cyber warfare settings, adversary modelling, attack modelling, defence modelling and data privacy in ML models. In this comprehensive survey, we integrate the various adversarial machine learning techniques in the cyber warfare setting to analyse the dynamic attack and defence strategies to improve the security of the simulated system.

REFERENCES

1. Huang, L.; Joseph, A.D.; Nelson, B.; Rubinstein, Benjamin I.P. & Tygar, J.D. Adversarial machine learning. Chicago, Illinois, USA. AISec' 11, October 21, 2011, pp. 43-58.
2. Goodfellow, I.J.; Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples. *In* the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 2015.
3. Kurakin, A.; Goodfellow, I.J. & Bengio, S. Adversarial examples in the physical world. *In* the International Conference on Learning Representations (ICLR), Toulon,

France, 2017.
4. Kurakin, A.; Goodfellow, I.J. & Bengio, S. Adversarial machine learning at scale. *In* the International Conference on Learning Representations (ICLR), Toulon, France, 2017.
5. Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L. & Yuille, A. Adversarial examples for semantic segmentation and object detection. arXiv:1703.08603v3 [cs.CV].
6. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B. & Swami, A. The limitations of deep learning in adversarial settings. *In* IEEE European Symposium on Security and Privacy (EuroS&P), 2016, pp. 372-387. doi: 10.1109/EuroSP.2016.36
7. Carlini, N. & Wagner, D. Towards evaluating the robustness of neural networks. *In* IEEE Symposium on Security and Privacy, San Jose, CA, USA, 2017, pp. 39-57. doi: 10.1109/SP.2-17.49
8. Biggio, B.; Nelson, B. & Laskov, P. Support vector machines under adversarial label noise. *In* JMLR: Workshop and Conference Proceeding, Taoyuan, Taiwan, 2011, pp. 97-112.
9. Srndic, N. & Laskov, P. Practical evasion of a learning-based classifier: A case study. *In* IEEE Symposium on Security and Privacy, San Jose, CA, USA, 2014, pp. 197-211. doi: 10.1109/SP.2014.20.
10. Xu, W.; Qi, Y.; Evans, D. Automatically evading classifiers: A case study on PDF malware classifiers. *In* Network and Distributed System Security Symposium 2016 (NDSS), San Diego, February 2016.
11. Khorshidpour, Z.; Hashemi, S. & Hamzeh, A. Learning a secure classifier against evasion attack. *In* IEEE 16th International Conference on Data Mining Workshop, Barcelona, Spain, 2016, pp. 295-302. doi: 10.1109/ICDMW.2016.0049
12. Liang B.; Li, H.; Su, H.; Bian, M.; Li, M. & Shi, X. Deep text classification can be fooled. arxiv:1704.08006 [cs. CR].
13. Kloft, M. & Laskov, P. Online anomaly detection under adversarial impact. *In* International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, Italy, 2010.
14. Biggio, B.; Didaci, L.; Fumera, G. & Roli, F. Poisoning attacks to compromise face templates. *In* International Conference on Biometrics (ICB), Madrid, Spain, 2013, pp. 1-7. doi: 10.1109/ICB.2013.6613006.
15. Mozaffari-Kermani, M.; Sur-Kolay, S.; Raghunathan, A. & Jha, N.K.; Systematic poisoning attacks on and defenses for machine learning in healthcare. *J. Biom. Health Infor*., 2015, **19**(6), 1893-1905. doi: 10.1109/JBHI.2014.2344095
16. Tramer, F.; Zhang, F.; Juels, A.; Reiter, M.K & Ristenpart, T. Stealing machine learning models via prediction APIs. *In* Proceedings of 25th Usenix Security Symposium, Austin, Texas, 2016.
17. Frekrikson, M.; Jha, S. & Ristenpart, T. Model inversion attacks that exploit confidence information and Basic Countermeasures. *In* Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15), Colorado, USA, 2015, pp. 1322-1333. doi: 10.1145/2810103.2813677.
18. Tramèr, F.; Papernot, N.; Goodfellow, I.J.; Boneh, D. & McDaniel, P. The space of transferable adversarial examples. arXiv:1704.03453v2[stat.ML].
19. Papernot, N.; McDaniel, P. & Goodfellow, I.J. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. arXiv: 1605.07277v1 [cs.CR].
20. Papernot, N., McDaniel, P., Goodfellow, I.J., Jha, S., Berkay Celik, Z. & Swami, A. Practical black-box attacks against machine learning. *In* Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS'17), Abu Dhabi, 2017, pp.506-519. doi: 10.1145/3052973.3053009.
21. Liu, Y.; Chen, X.; Liu, C. & Song, D. Delving into transferable adversarial examples and black-box attacks. arXiv:1611.02770v3 [cs.LG].
22. Hayes, J. & Danezis, G. Machine learning as an adversarial service: Learning black-box adversarial examples. arXiv: 1708.05207v1 [cs.CR].
23. Shokri, R.; Stronati, M.; Song, C. & Shmatikov, V. Membership inference attacks against machine learning models. *In* IEEE Symposium on Security and Privacy (S&P) -- Oakland, 2017, pp. 3-18. doi: 10.1109/SP.2017.41
24. Biggio, B.; Fumera, G. & Roli, F. Adversarial pattern classification using multiple classifiers and randomisation. *In* Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition (SSPR'08), Florida, USA, 2008, pp. 500-509. doi:10.1007/978-3-540-89689-0_54
25. Biggio, B.; Fumera, G. & Roli, F. Multiple classifier systems under attack. *In* Proceedings of the 9th international conference on Multiple Classifier Systems, Cairo, Egypt, 2010, pp. 74-83. doi: 10.1007/978-3-642-12127-2_8.
26. Biggio, B.; Fumera, G. & Roli, F. Security evaluation of pattern classifiers under attack. *In* IEEE Transactions on Knowledge and Data Engineering, 2014, **26**(4), pp. 984-996.
27. Lowd, D. & Meek, C. Adversarial learning. KDD, Illinois, USA, 2005, pp. 641-647.
28. Grobhans, M.; Sawade, C.; Bruckner, M. & Scheffer, T. Bayesian games for adversarial regression problems. *In* Proceedings of the 30th International Conference on International Conference on Machine Learning, Atlanta, GA, USA, 2013, pp. 55-63.
29. Biggio, B.; Nelson, B. & Laskov, P. Poisoning attacks against support vector machine. ICML, Edinburg, Scotland, 2012, pp. 1467-1474.
30. Xiao, H.; Xiao, H. & Eckert, C. adversarial label flips attack on support vector machines. *In* ECAI'12 Proceedings of

the 20th European Conference on Artificial Intelligence, Montpellier, France, 2012, pp. 870-875.
doi: 10.3233/978-1-61499-098-7-870

31. Xiao, H.; Biggio, B.; Nelson, B.; Xiao, H.; Eckert, C. & Roli, F. Support vector machines under adversarial label contamination. *In* Neurocomputing, 2014, **160**(C), 53-62.
doi: 10.1016/j.neucom.2014.08.081
32. Burkard, C. & Lagesse, B. Analysis of causative attacks against SVMs learning from data streams. IWSPA, Scottsdale, Arizona, 2017, pp. 31-36.
doi: 10.1145/3041008.3041012
33. Szegedy, C.; Erhan, D.; Ilya Sutskever, W.Z.; Goodfellow, I.J.; Bruna, J. & Fergus, R. Intriguing properties of neural networks. arXiv: 1312.6199v4 [cs.CV].
34. Gu, T.; Dolan-Gavitt, B. & Garg, S. BadNets: Identifying vulnerabilities in the machine learning model supply chain. arXiv:1708.06733v1 [cs.CR].
35. Dezfooli, S. M.; Fawzi, A. & Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks. *In* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2574-2582.
doi: 10.1109/CVPR.2016.282.
36. Munoz-Gonzalez, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E. C. & Roli, F. Towards poisoning of deep learning algorithms with back-gradient optimization. *In* Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 27-38. doi:10.1145/3128572.3140451.
37. Narodytska, N. & Kasiviswanathan, S. Simple black-box adversarial attacks on deep neural networks. *In* IEEE Conference on Computer Vision and Pattern Recognition Workshop, Hawaii, USA, 2017, pp. 1310-1318.
doi: 10.1109/CVPRW.2017.172
38. Papernot, N.; McDaniel, P.; Swami, A. & Harang, R. Crafting adversarial input sequences for recurrent neural networks. *In* Military Communications Conference, MILCOM, LA, USA, 2016.
39. Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. & Bengio, Y. Generative adversarial nets. *In* NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, Canada, 2014, pp. 2672-2680.
40. Hu, W. & Tan, Y. Generating adversarial malware examples for black-box attacks based on GAN. arXiv:1702.05983v1 [cs.LG].
41. Shen, S.; Jin, G. & Gao, K. APEGAN: Adversarial perturbation elimination with GAN. arXiv: 1707.05474v3 [cs.CV].
42. Kos, J.; Fischer, I. & Song, D.; Adversarial examples for generative models. arXiv:1702.06832v1 [stat.ML].
43. Tabacof, P.; Tavares, J. & Valle, E. Adversarial images for variational autoencoders. arXiv:1612.00155v1 [cs.NE].
44. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.J. & Frey, B. Adversarial autoencoders. arXiv:1511.05644v2 [cs.LG].
45. Biggio, B.; Pillai, I.; Rota Bulo, S.; Ariu, D.; Pelillo, M. & Roli, F. Is data clustering in adversarial settings secure?. AISec, Berlin, Germany, 2013, pp. 87-98.
doi: 10.1145/2517312.2517321
46. Dutrisac, J.D. & Skillicorn, D.B. Hiding clusters in adversarial settings. *In* IEEE International Conference on Intelligence and Security Informatics(ISI), 2008, pp. 185-197.
doi: 10.1109/ISI.2008.4565051
47. Biggio, B. Poisoning complete-linkage hierarchical clustering. *In* Joint IAPR Int'l Workshop on Structural, Syntactic, and Statistical Pattern Recognition (LNCS), Joensuu, Finland, 2014, **8621**, pp. 42-52.
doi: 10.1007/978-3-662-44415-3_5
48. Biggio, B. Poisoning behavioral malware clustering. *In* Proceedings of the 2014 ACM Workshop on Artificial Intelligence and Security, colocated with CCS '14, Scottsdale, Arizona, USA, 2014, pp. 27-36.
doi: 10.1145/2666652.2666666
49. Uther, W. & Veloso, M. Adversarial reinforcement learning. 1997.
50. Behzadan, V. & Munir, A. Vulnerability of deep reinforcement learning to policy induction attacks. *In* International Conference on Machine Learning and Data Mining in Pattern Recognition, 2017.
51. Huang, S.; Papernot, N.; Goodfellow, I.J; Duan, Y. & Abbeel, P. Adversarial attacks on neural network policies. arXiv: 1702.02284v1 [cs.LG].
52. Kos, J. & Song, D. Delving into adversarial attacks on deep policies. Workshop track - ICLR, 2017.
53. Chen Lin, Y.; Wei Hong, Z.; Hong Liao, Y.; Shih, M.; Liu, M. & Sun, M.; Tactics of adversarial attack on deep reinforcement learning agents. *In* Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, Melbourne, Australia, 2017.
54. Xie, P. CryptoNets: Neural networks over encrypted data. arXiv:1412.6181 [cs.LG].
55. Dowlin, N. CryptoNets: Applying neural networks to encrypted data with high thorughput and accuracy. *In* Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016, pp. 201-210.
56. Rouhani, B.D.; Sadegh Riazi, M. & Koushanfar, F. DeepSecure: Scalable provably-secure deep learning. arXiv:1705.08963 [cs.CR].
57. Ohrimenko, O.; Schuster, F.; Fournet, C.; Mehta, A; Nowozin, S.; Vaswani, K. & Costa, M. Oblivious multi-party machine learning on trusted processors. *In* 25th USENIX Security Symposium, Austin, TX, USA, 2016, pp. 619-636.
58. Mohassel, P. & Zhang, Y. SecureML: A system for scalable privacy-preserving machine learning. *In* IEEE Security and Privacy Symposium, San Jose, CA, USA, 2015, pp. 19-38.
doi: 10.1109/SP.2017.12
59. Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.J. & Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *In* International Conference on Learning Representations (ICLR), Toulon,

France, 2017.
60. Shokri, R. & Shmatikov, V. Privacy-preserving deep learning. CCS'15, Colarado, USA, 2015, pp. 1310-1321. doi: 10.1145/2810103.2813687
61. Abadi, M.; McMahan, H.B.; Chu, A.; Mironov, I.; Zhang, L.; Goodfellow, I.J. & Talwar, K. Deep learning with differential privacy. CCS'16, Vienna, Austria, 2016, pp. 308-318. doi: 10.1145/2976749.2978318
62. Papernot, N.; McDaniel, P.; Sinha, A. & Wellman, M. Towards the science of security and privacy in machine learning. arxiv:1611.03814.
63. Hinton, G.; Vinyals, O. & Dean, J.; Distilling the knowledge in a neural network. arXiv:1503.02531v1 [stat.ML].
64. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S. & Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. *In* the 37th IEEE Symposium on Security & Privacy, San Jose, CA, USA, 2016, pp. 582-597. doi: 10.1109/SP.2016.41
65. Carlini, N. & Wagner, D. Defensive distillation is not robust to adversarial examples. arXiv, 2016.
66. Papernot, N. & McDaniel, P. Extending defensive distillation. arxiv: 1705.05264v1 [cs:LG].
67. Tramer, F.; Kurakin, A.; Papernot, N.; Boneh, D. & McDaniel, P. Ensemble adversarial training. arXiv:1705.07204v2 [stat.ML].
68. Miyato, T.; Dai, A. M. & Goodfellow, I.J. Adversarial training methods for semi-supervised text classification. *In* International Conference on Learning Representations (ICLR), Toulon, France, 2017.
69. Xu, W.; Evans, D. & Qi, Y. Feature squeezing: detecting adversarial examples in deep neural networks. arXiv:1704.01155v1 [cs.CV].
70. Lu, J.; Issaranon, T. & Forsyth, D. SafetyNet: Detecting and rejecting adversarial examples robustly. arXiv:1704.00103v2 [cs.CV].
71. Metzen, J.K.; Genewein, T.; Fischer, V. & Bischoff, B. On Detecting adversarial perturbations. ICLR, Toulon, France, 2017.
72. Barreno, M.; Nelson, B.; Joseph, A.D. & Tygar, J.D. The security of machine learning. *Machine Learning J.*, 2010, **81**(2), 121-148. doi: 10.1007/s10994-010-5188-5
73. Bhagoji, A.N; Cullina, D.; Sitawarin, B. & Mittal, P. Enhancing robustness of machine learning systems via data transformations. arxiv:1704.02654v3 [cs:CR].
74. Athalye, A. & Sutskever, I. Synthesizing robust adversarial examples. arXiv:1707.07397v1 [cs.CV].
75. Demontis, A.; Melis, M.; Biggio, B.; Maiorca, D.; Arp, D.; Rieck, K.; Corona, I.; Giacinto, G. & Roli, F. Yes. Machine learning can be more secure! A case study on android malware detection. *In* IEEE Transactions on Dependable and Secure Computing, 2017, Early Access, pp 1-1. doi: 10.1109/TDSC.2017.2700270
76. Xiao, H.; Biggio, B.; Brown, G.; Fumera, G.; Eckert, C. & Roli, F. Is Feature selection secure against training data poisoning? *In* the Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015, **37**, pp. 1689-1698.
77. Zhang, F.; Chan, P.; Biggio, B.; Yeung, D.S. & Roli, F. Adversarial feature selection against evasion attacks. *IEEE Trans. Cybernetics*, 2016, **46**(3), 766-777. doi: 10.1109/TCYB.2015.2415032
78. Ravishankar, M.; Vijay Rao, D. & Kumar, C.R.S. Game theory based defence mechanisms of cyber warfare. *In* 1st Conference on Latest Advances in Machine Learning and Data Science LAMDA, NIT Goa, 2017.
79. Ravishankar, M.; Vijay Rao, D. & Kumar, C.R.S. A Game theoretic approach to modeling jamming attacks, In delay tolerant networks. *Def. Sci. J.*, 2017, **67**(3), 282-290. doi: 10.14429/dsj.67.10051
80. Ravishankar, M.; Vijay Rao, D. & Kumar, C.R.S. A game theoretic software test-bed for cyber security of critical infrastructure. *Def. Sci. J.*, 2018, **68**(1), 54-63. doi: 10.14429/dsj.68.11402
81. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G & Roli, F. Evasion attacks against machine learning at test time. *In* Lecture Notes in Computer Science, 2013, **8190**, pp. 387-402. doi: 10.1007/978-3-642-40994-3_25

ACKNOWLEDGEMENTS

The author would like to thank Dr A.K. Sinha, Scientist G, DRDO-Defence Terrain Research Laboratory, Delhi and Dr D. Vijay Rao, Scientist G, DRDO-Institute for Systems Studies and Analyses, Delhi for the fruitful discussions, encouragement and guidance; and the anonymous reviewers for their suggestions and critical reviews that have greatly improved the quality of the paper.

CONTRIBUTOR

Mr Vasisht Duddu is pursuing BTech (Electronics and Communications Engineering) from Indraprastha Institute of Information Technology (IIIT), Delhi and is currently working as a researcher at System Security Lab, School of Computing, National University of Singapore(NUS), Singapore. His primary areas of research are security, privacy, anonymity and applied cryptography.