Safeguarding Machine Learning Systems: A Comprehensive Analysis of Security Concerns and Defensive Strategies

A technical paper prepared for presentation at SCTE TechExpo24

Shivam Gupta
Security Engineer
Cantata Health Solutions
sg311098@gmail.com

Table of Contents

List of Figures

List of Tables

1. Introduction

Machine Learning (ML) systems have made major advancements in recent years and are constantly used in a wide range of applications like image processing, autonomous cars, speech and gesture recognition, credit card fraud detection, and smart healthcare, to name a few. There are hardly any areas of business where ML has not been applied. Due to this range of applications and the accuracy of the ML systems, millions of dollars are being invested by private and government organizations across the globe [1]. The data collected by mobile devices and systems, universities, banks, corporate organizations, and even in our homes, which might be private or public is being used by these Machine Learning applications. Sometimes private data needs to be stored in centralized locations in plain text for the algorithms to extract the feature or pattern and to build a model of that application using Machine Learning systems. The associated threats are not only limited to the leakage of this private data to an insider of that organization or an outsider eavesdropping on the private data. In addition to this there is a possibility of extracting other confidential information about an individual or a whole company's data even if the data is anonymized by methods like data masking, pseudonymization, or the dataset itself, and the model would not be accessible and result revealing the final results [1].

The history of security mechanisms has shown that threat detection is like playing a cat-and-mouse game. With every new malware detection method, there is always a new evasion technique in attackers 'minds. Backdoor and code injection methods were invented by intruders to evade behavior detection. Also, when signature-based detection methods were introduced by defenders, attackers started using packers, compressors, and polymorphism to bypass it, making the systems confused [2]. At the moment, where Machine Learning has started being used as a security solution for many issues, cybercriminals have already started to trick them. Al-Rubaie et al. [2] list some of the attribute reasons to these attacks:

ML is making tremendous advancement in significant areas such as healthcare, finance, public sector, and defense, that exchange very sensitive data.

Complexity and inconsistency concerns are rising as a huge number of devices are connecting and using gigantic datasets for training and testing [2].

Being an evolving field, many industries are using numerous applications of ML without considering the security associated with this system in mind. This results in an increased number of security threats related to the Confidentiality, Integrity, and Availability (CIA) triad.

Some security computations use a significant amount of computing resources. Because of the limited capabilities of ML systems, many of them lack encryption. This absence of encryption across ML systems leaves the gate open to be discovered and exploited by intruders [2].

Familiarity with the applications and requirements of machine learning in diverse areas is not enough, however. We need to see the other side of ML Systems, which is identified by the intruders and attackers to compromise these systems for different reasons.

This study discusses various Machine Learning functionalities and applications, and it additionally covers the possible threats associated with the existing methods of gathering data and developing Machine Learning Systems. The paper further detail security measures to prevent ML systems from these threats/attacks of individuals or organizations. The motivation is to fill up the gap between ML systems and the associated threats with its privacy and security by making the individuals more aware of the potential threats, the preventive solutions, and the mitigation techniques.

2. Machine Learning

Arthur Lee Samuel, a pioneer in the field of computer gaming and artificial intelligence, described machine learning as "a field of study that gives computers an ability to learn without being explicitly programmed" [22]. ML systems are used to understand the performance of several tasks that generalize with the data. These tasks possibly provide highly accurate predictions or find the pattern in the data precisely [2].

The training data that is introduced to the ML system is represented as a set of several data samples. For instance, to form a feature vector, which is none more than a combination of 10,000 vectors, which are formed with the photo pixels (100x100) that are represented by a grayscale matrix (0-225). These pictures which are represented as a feature vector are generally labeled with some information like the name of the person, date, and time of photo. The ML algorithm trains the model with the dataset of numerous feature vectors, and the associated labels of each vector to develop a machine learning model. Using these datasets as input and training a model with this dataset is called the "Training/ Learning" phase. After training with the appropriate data, the model should be able to provide the predicted results in its testing phase. The accuracy of the model after the testing phase determines how well the ML model generalized to unseen data. Predicted results are measured based on trial and error in some fixed size epochs and sometimes depend upon the data properties (data quality and quantity) [2].

Generally, in some applications like feature extraction, it is important to get some useful features from the raw data, to be precise pre-processing data (of the pictures, taken as an example), then applying some image processing techniques (such as image cropping and resizing to required pixels size, 100x100) to make it useful input for ML system [1][2]. These are applied based on the applications and learning of the ML system and where it is applied to. We can classify ML systems based on their learning type into 'supervised 'or 'unsupervised', or the blend of both:

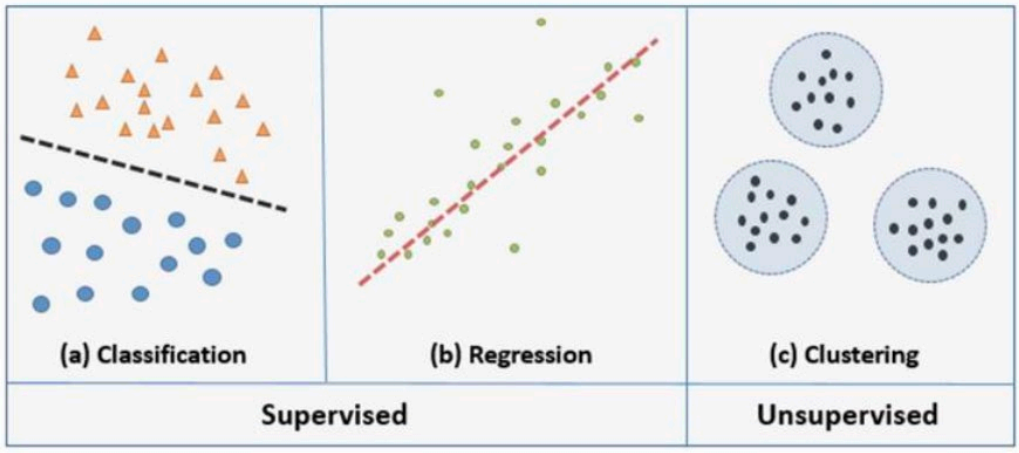

Figure 1 - Classification: finding a separating dashed line (b) Regression: fitting a predictive; (c) Clustering

Supervised Learning: In supervised learning, we train or test the ML machine by providing data that is well labeled with class (Classification) or a continuous stretch of real values (Regression). These labeled data can be used to develop the models and then predict the labels of new feature vectors.

In *Classification*, the samples are distributed between two or more classes, and the ML system is used to determine the class in which the new sample must belong. As we can see in Figure – 1(a), algorithms may classify the samples by dividing them with a hyperplane. For example, in a face recognition model, a face image can be tested by classification based on the face features to determine which class it should go into.

Various classification algorithms can be used for supervised applications such as Naive Bayes Classifiers, K-NN (k-nearest neighbors), or Decision Trees [2].

Regression is when the label of a sample is a continuous stretch of values (which is also called the response variable) rather than a discrete independent value or features [2]. A regression model aims to fit a model for the prediction of observed samples to minimize the distance between observed data and predictive model (a line), as shown in the Figure – 1(b). A perfect example of regression would be predicting the value of a house in dollars to sell, perhaps in some range.

Unsupervised Learning: This type of method deals with unlabeled data as a feature vector, which does not comprise labels of class or a response variable. The objective of this learning is to find a pattern or structure of the sample.

Clustering is the most common and widely used type of unsupervised learning in which clusters are formed according to their properties (Figure – 1c). Some clustering methods include Hierarchical clustering, K-means clustering, and Independent Principal Component Analysis [2]. Some applications are not restricted to either supervised or unsupervised ML learning. These include Dimensionality reduction and Recommender Systems.

3. Security Requirements

Machine Learning is a capability that increases our convenience from YouTube's recommendations based on the previous search to filtering spam and phishing emails. ML is an imaginative resource of advanced technology, but it is always surrounded by uninvited threats and attacks. Every business between the company and clients develops a level of trust. This trust remains in place if a company makes every effort to keep the customer's data un-compromised and privacy is maintained. Important requirements that must be satisfied when training and testing ML systems 'security and privacy are listed below [3].

3.1. Data Confidentiality and Availability

With Machine Learning systems, confidentiality is defined concerning the data inserted and the model used to process the data.

Attacks are being performed on confidentiality to expose the sensitive data used for training and testing (e.g., bank transactions, healthcare data) and the model structure (that is equally important intellectual property). Availability of resources is the highest priority for ML systems. That is how they can predict the sample, but some adversarial behavior tries to prevent legitimate users from accessing meaningful results or other feature vectors of the model itself, like DoS or DDoS attack, so to resist these types of attacks that can affect the availability should be taken proper measure of [3].

3.2. Privacy

Another security requirement is privacy. Attacks might affect the privacy of the data used by the model, especially when the users are not trustworthy. For example, bank account data or healthcare patient data

used to train diagnosis devices is very sensitive, and needs to be secure from unauthorized users, and even if intruders get the data, they should not be able to get something meaningful from that [3].

3.3. Integrity

As the data used for the model is sensitive in nature, so it should be protected against alteration. In other words, the data should be tamper-proof during the training and testing phase, as this will maintain the integrity [3]. An attack on integrity is seen in the outputs or in the training prediction. This involves the modification or manipulation of data that might affect the performance of the model. So, it is better to make sure that integrity is not being compromised.

4. Attack Taxonomy

This section talks about the threats and attacks that are handled by ML systems. Figure 2 shows all possible threats along the process of machine learning.

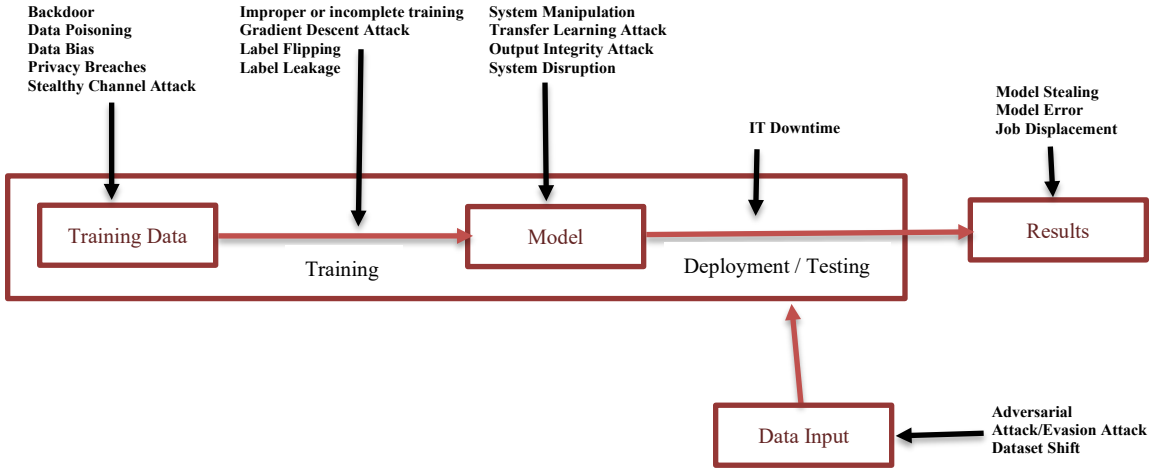

Figure 2 - Intruder attacks on different phases of Machine Learning System

Threats to machine learning can be divided into two main categories based on the section where the attack has been done. These are Threats to the training data, and Threats on the Algorithm or Model. These are be further sub-classified based on the phase when they are attacked in the life cycle of the ML algorithm, which is categorized as “before or during the training phase” and “threats after the training” of ML system. Figure - 2 shows some of the attacks that are possible in the machine learning model in different phases [4].

4.1. On Data

Attacks may be executed on the data for training and testing the model. This data is of the highest priority, as the model will predict the accuracy of the algorithm based on reality and originality of the data.

4.1.1. Before Training

4.1.1.1. Stealthy Channel Attack

To develop a high-quality machine learning model, data quality is the major factor. Therefore, it is important to collect useful and relevant data from a trusted source, as collecting data from different non-trusted sources might compromise the system. This is where intruders can insert or modify data that can lead to inaccuracy and sometimes even crash an ML system. This attack is known as a Stealth channel attack. This is a phase before the model training where collected data should be checked and examined before entering it into the machine learning system [5].

4.1.1.2. Data Poisoning

The most common and efficient attack on a machine learning system is data poisoning. As we have seen in Stealthy Channel Attack, data is a crucial part of a ML model, so even a small change can render the system unusable. This type of attack is quite similar to the Stealthy Channel Attack, where an attacker tries to use an ML system vulnerability and try to manipulate data to be used for the training phase [1]. Data poisoning is directly responsible for two aspects of data, Data Confidentiality and Data Trustworthiness.

Many ML systems are used for healthcare, finance, and banks, and these contain highly confidential and private information, which needs to be confidential [5]. If an attacker performs a data poisoning attack, then this confidentiality is lost. Maintaining the confidentiality of data is the most challenging task of any ML system, and this is one factor that shows how secure and good a system is. This is not much different from data trustworthiness. It is a loss of confidence in the confidentiality of the data and the lack of trust in ML systems can be combinedly referred to as data poisoning [5].

4.1.1.3. Backdoor

Recent studies show that an attacker can hide a backdoor that will trigger if some specific condition arrives, either in the training phase, or after the pre-training of the model. This backdoor might not directly affect the model, and the model seems to work normally with the stealthy functioning of backdoor, but if it gets executed then we cannot predict the consequences of the attack as shown in Figure - 3 [1].

Chen et al. [6] proposed a backdoor attack on ML models by using data poisoning. More precisely, poisoning samples are inserted into the training dataset to embed a backdoor. This attack can work for a weak model as well. In other words, it does not require knowledge of the model used or the training set. In this research, only 50 poisoning samples are injected in the training data, and the attack

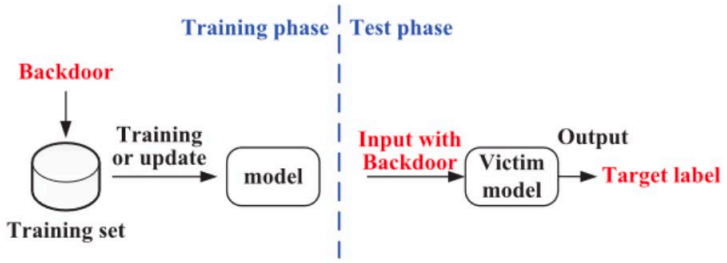

Figure 3 – Overview of Backdoor Attack

success rate was above 90% [6]. Bagdasaryan et al. [7] demonstrate backdoor attacks on Federated Learning, which is believed to be a secure privacy-preserving learning framework. They showed that malicious data can create a stealthy backdoor function into the federated model using model replacement.

In both methods, the attackers first insert the required backdoor into the data poison it and then inject this poisoned data into the training model to re-train the target model [6][7]. These methods are silent and can perform the backdoor without affecting the performance of the model. The accuracy fluctuates by only 1-2%.

4.1.2. After Training

4.1.2.1. Adversarial Examples / Evasion Attack

Evasion Attack is another important and highly efficient security threat for ML systems. This attack involves continuous examining of classifiers by the attacker with new inputs in order to evade detection, hence sometimes these types of attacks are also called "adversarial inputs", since they are developed to bypass the classifiers [1]. Let us consider an image of a panda and how it evades the system identification or impersonates others. The attack is performed by adding a small perturbation that has already been calculated by the attacker to make the algorithm identify the image as accurate with high confidence, as shown in Figure - 4.

This resulted in the system recognizing an output image of a gibbon with high accuracy i.e., 93.3% [4].

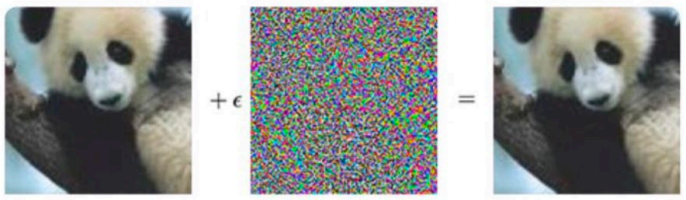

Figure 4 - Original and Perturbed Image

Another example of an evasion attack involves building a malicious document to evade spam filters, where an intruder observed that Gmail displays only the last attachment if the same multi-part attachment appears multiple times in the email [4]. Therefore, attackers use this vulnerability by adding an invisible multipart attachment that contains many reputable domains to evade recognition.

4.2. On Model / Algorithm

4.2.1. Before Training

4.2.1.1. Gradient Descent Attack

A machine learning model tries to learn and train itself by the trial-and-error method. In the first epoch, it is highly difficult to predict the results accurately. Generally, the model uses actual values to evaluate the predicted values, and as the number of epochs increases, the model tries to descend toward the expected value by adjusting a constant variable. This process of getting near to the actual results is called gradient descent. A gradient descent attack is undertaken while the model is in the training phase. In gradient descent, the model continues to iterate itself by tuning the constant variable until it is confident that the results are at high accuracy [5].

Gradient descent attacks can be done mainly in two ways. First, the model can be driven into an infinite loop of iteration by making it obvious that the current epoch is still not close to the expected value. This can be achieving by changing the expected value continuously to confuse the model at every epoch. Hence it goes into an infinite loop finding the actual value, and training never comes to an end result.

Secondly, an attacker can make the model believe that it has attained the desired value, which was expected by the model after training, and the model is mistakenly made to believe that the predicted value is the expected/actual value [5]. This attack makes it difficult to train the model accurately, and due to this incomplete training, it is highly probable to notice the inaccuracy of which is nothing but the compromised system.

4.2.1.2. Label Flipping

In the Label Flipping attack, data is poisoned by modifying the data label. The training data inserted in the ML system includes the combination of expected/output result and the given input, generally in Supervised learning. These expected outputs may be of the same or different group, if these are of a distinct group, then called labels. In a label flipping attack, the attacker makes these labels interchanged with each other [5].

In the below example, consider two tables. The first table is the original data, and the second table shows the data after label flipping [5].

Table 1 - Original Data

Cities	Country
Los Angeles	United States
New Delhi	India
Mumbai	India
Chicago	United States
Bangalore	India

Table 2 - Data after Label Flipping

Cities	Country
Los Angeles	India
New Delhi	United States
Mumbai	United States
Chicago	India
Bangalore	United States

The table shows the Input data (Cities) with the associated Labels (Country). Table 1 shows the original data, where there is a correct relation between the input data and labels, but in Table 2, it gets altered [5].

4.2.2. After Training

4.2.2.1. System Manipulation

Machine learning systems never stop learning, they continue learning and enhancing themselves. This enhancement is done by taking continuous feedback from the data and environment, which is alike to reinforcement models which take constant feedback from an element. This is an attack where attackers attempt to steer the system in the wrong direction by providing some false data as feedback to the system [1].

After 'n' number of iteration (Epochs) model performance starts degrading instead of improving accuracy, thus shifting the behavior of the system and making the system useless [5].

4.2.2.2. Transfer Learning Attack

Sometimes a company needs to use pre-trained machine learning models. One reason can be the high amount of training data, which takes a great deal of time to train.

These applications require a huge number of computational resources. Hence pre-trained models are preferred [4]. These pre-trained models are tweaked and fine-tuned according to the application's use and its requirements. But as the model has been pre-trained, there is no guarantee that the model is trained on the advertised dataset [5]. This loophole can be exploited by the attacker who might modify or replace the original model with a malicious one [5].

4.2.2.3. Output Integrity Attack

If the intruder makes it between the model and the interface used to display the result, then the modified (by the attacker) results can be shown. This attack is known as the Output Integrity Attack [5]. Due to the lack of knowledge about how ML system internals work theoretically, it becomes hard to predict the actual results. Hence, the output is taken at face value. This is where attackers exploit and ultimately compromise the integrity of the model [5].

4.2.2.4. Model Stealing / Extraction

Recent findings show that an attacker can steal the ML model by observing the labels and confidence levels with respect to the assigned inputs. This attack is known as Model Stealing, also called Model

Extraction, which has become an emergent threat [1]. Generally, it is applied after the training phase at the time of expected results extraction.

Tramer et al. [8] developed the first model stealing attack, i.e., an attacker makes all possible ways to steal the ML model through numerous user inquiries. When inserting and processing normal queries through prediction APIs, the model returns a predicted label with a confidence level associated with that. Based on these services, the author showed the model stealing attacks on three types of models:

Logistic regression, Decision trees, and Neural Networks [1][8]. The ML services that are used for the evaluations are Amazon and BigML. Yi et al. [9] proposed a method of model extraction by building a functionally equivalent model based on machine learning. This method works in a black-box setup, where the attacker gets all the predicted labels from the target model and uses the ML system to imply and build a comparable model [9]. More precisely, they use the input data to query the target model and use the output data for their model as the labels to train a new model that has similar functions.

These methods [8][9] train a model similar to the target model using the black-box technique, which does not need the attacker to have knowledge about the target system. But they need to query the target system multiple times to predict the output data precisely. If the target system limits the number of queries, a model stealing attack is not possible.

5. Defensive Procedures

5.1. Defending Against Poinsoning

To improve the robustness of machine learning algorithms and mitigate the impact of outliers on the trained model focusing on binary classification problems, Biggio et al. [10] proposed a model by considering poisoning attack mitigation as an outlier detection problem. These are few in number but have shifted the distribution as compared to the conventional training sample set. Therefore, researchers used Bagging Classifiers, which is a perfect model to decrease the effects of outliers (poisoning samples) from the training dataset. More precisely, they have used different data sets to train the model each time, and after repeating iterations several times, they predicted the results combining all the predictions from different datasets on the classifier to reduce the influence of outliers in the training set [10]. This helped in the application of Spam-filtering and web-based Intrusion Detection Systems (IDS) against poisoning attacks [10].

For defending healthcare systems, Mozaffari-Kermani et al. [11] proposed a method where he monitored the accuracy deviation of training data and the additional data into the dataset. This technique is generic but provides protection against poisoning attacks for different target models [11]. However, this model is computationally intensive, as it requires retraining the model periodically.

5.2. Defending Againt Backdoor

Chen et al. [14] proposed the Activation Clustering (AC) method for protecting Machine Learning systems by identifying the poisonous training samples in the training data and removing backdoors from the model. This model first analyses the model activations of the training samples and determines whether the sample is poisoned, and, if so, which segment of data is poisoned. Thus, it detects all poisonous backdoor samples even with various backdoor formations. Liu et al. [15] examined two security measures to protect the machine learning model from backdoor attacks, which are pruning and fine-tuning. Pruning helps in reducing the size of the backdoored system by decreasing neurons that are hidden on clean inputs, therefore deactivating backdoor components. Following pruning, fine-tuning is implemented to

defend against a strong attack which is capable of breaking pruning. Fine-tuning is a small amount of retraining on a fresh clean sample [15]. As fine-tuning provides a high degree of protection against backdoors, we prefer a combination of pruning and fine-tuning termed as fine pruning, as it is the most efficient in disabling backdoor attacks [14][15]. These methods are suitable for most of the machine learning models, but they require high computational expenses to detect and disable backdoors.

5.3. Defending Against Adversarial Examples

The most effective method of defense against adversarial examples is to increase the adversarial examples, detect them, train those examples and finally apply defensive distillation. The target model attempts to add more noise to create effective adversarial examples [1]. According to researchers, evasion attacks are relatively hard to detect, as it is unclear, and also it is hard to manage the testing sample, which is used to predict the adversarial example. Some detection techniques are efficient, while some are inefficient.

Therefore, it is better to mark the labels of the test examples. For instance, in an autonomous self-driving car, it marks all the labels itself to detect adversarial examples. Meng and Chen [12] on the other hand argue and state that if during the verification of the adversarial example, it is proved through testing example, then adding labels is not required for classifiers. After detecting adversarial examples, training is required. Goodfellow et al. [13] states that the method to train the model is through expanding training data with several adversarial examples and refers to it as adversarial training. To manage evasion attacks, benign training examples are matched against adversarial training examples. The learner/user will be able to trace back the algorithm to understand the Machine Learning System through the original benign example and the attack adversarial example [13]. Finally, distillation is applied. For each training example, the model produces a set of confidence levels. These levels are treated as a mark for the training example. So, reading these labels and confidence levels, the model can differentiate original and evasion attack data.

5.4. Defending Againt Model Stealing

An instinctive approach against a model stealing attack is when the label is outputted without giving the confidence information. This might degrade the performance of the service, but it is a secure method to prevent theft. Lee et al. [16] proposed a method to protect machine learning systems by injecting false perturbations in the confidence information to deceive the adversary. This results in the adversary only left with the labels to steal the model. Moreover it will require numerous different queries to extract the model [16]. This is an effective method to protect the model from extraction attacks, but if the adversary successfully gets enough queries, they might still be able to steal the model.

Another technique is proposed by Juuti et al. [17] to detect model stealing attacks, named PRADA. Since the attacker steals the model through APIs, PRADA analyses the distribution of queries and detects a continuous set of unusual queries. This is a common but effective method, but it does not provide robustness for the dummy queries [18], which helps the attacker to make some anomalous queries invisible.

5.5. Protecting Sensitive Data

The defense for machine learning systems against protection of sensitive data can be done by cryptographic primitive-based approaches, such as differential privacy and homomorphic encryption.

Abadi et al. [19] proposed a differential primitive-based Machine Learning system. They also demonstrated methods to enhance the efficiency of the differential primitive-based training, which improves compatibility between the privacy, efficiency, complexity of software, and the model quality. This method is efficient for the protection of sensitive data, but additional noise makes the model less accurate. Jayaraman et al. [21] also demonstrate that there is a connection between the privacy and the performance of the model in a differential primitive-based system [21]. More precisely, it says that when we protect the privacy of the model, differential primitive-based might sacrifice performance as compared to the original.

Phong et al. [20] state that the distributed learning model for privacy protection might be able to expose some secret information to the server. Therefore, they imposed a technique by applying asynchronous stochastic gradient descent to the machine learning model and introduce homomorphic encryption to the model [20]. The homomorphic-based encryption method uses cryptographic primitives to make sure that the security, privacy as well as accuracy is maintained, but this model imposes high computational overhead in the training phase of the algorithm.

6. Conclusion

Machine learning has integrated with crucial industrial services with many applications, yet machine learning systems still deal with a range of security threats throughout different phases. Machine learning security is the most active and important topic for research and study which is still an open problem. In this paper, we have presented a comprehensive review of some major security challenges that are currently being faced with the corresponding countermeasures.

A typical conclusion is that the threats are genuine, and new threats are continually emerging. As the data plays an important role for machine learning models, most of the threats target data, to alter, steal, or destroy dataset for that model. Another major target is the model itself. As we have seen in the Model Stealing attack, adversaries try to steal the model using some pre-trained model or by other means. Therefore, the privacy and integrity of data as well as the model are of the utmost importantance. Rather than focusing on one part of the model i.e., training or testing, we should consider all the phases of machine learning lifecycle and take all possible security measures to make those vulnerability free. This paper can positively provide comprehensive guidelines for developing secure, robust, and private machine learning systems.

Abbreviations

ML	machine learning
CIA	Confidentiality, Integrity and Availability
K-NN	k-nearest neighbor
DoS	denial of service
IDS	intrusion detection system
AC	activation clustering

Bibliography & References

[1] M. Xue, C. Yuan, H. Wu, Y. Zhang, and W. Liu, "Machine Learning Security: Threats, Countermeasures, and Evaluations," IEEE access, vol. 8, pp. 74720–74742, 2020, doi: 10.1109/ACCESS.2020.2987435.

[2] Al-Rubaie, Mohammad. Chang J. Morris, "Privacy Preserving Machine Learning: Threats and Solutions", IEEE Security and Privacy Magazine, 2018

[3] Papernot, Nicolas. McDaniel, Patrick. Sinha, Arunesh. Wellman, P. Michael., "SoK: Security and Privacy in Machine Learning", IEEE European Symposium on Security and Privacy 2018

[4] Machine Learning Security: 3 Risks To Be Aware Of, July 11, 2019, accessed December 01, 2021, <https://www.plugandplaytechcenter.com/resources/machine-learning-security-3-risks-be-aware/>

[5] P.N, Tatwadarshi, January 2021, Security Threats to Machine Learning Systems, Analytics Vidhya, accessed 15 October 2021, <https://www.analyticsvidhya.com/blog/2021/01/security-threats-to-machine-learning-systems/>

[6] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, ''Targeted backdoor attacks on deep learning systems using data poisoning, ''2017, arXiv:1712.05526.

[7] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, ''How to backdoor federated learning,'' 2018, arXiv:1807.00459.

[8] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, ''Stealing machine learning models via prediction APIs, ''in Proc. 25th USENIX Secur. Symp., Aug. 2016, pp. 601–618

[9] Y. Shi, Y. Sagduyu, and A. Grushin, ''How to steal a machine learning classifier with deep learning, '' in Proc. IEEE Int. Symp. Technol. Homeland Secure. (HST), Apr. 2017, pp. 1–5

[10] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, ''Bagging classifiers for fighting poisoning attacks in adversarial classification tasks, ''in Proc.10th Int. Conf. Mult. Classif. Syst., Jun. 2011, pp. 350–359.

[11] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, ‘‘Systematic poisoning attacks on and defenses for machine learning in healthcare, ’’IEEE J. Biomed. Health Informat., vol. 19, no. 6, pp. 1893–1905, Nov. 2015.

[12] D. Meng and H. Chen, ‘‘MagNet: A two-pronged defense against adversarial examples, ’’in Proc. ACM SIGSAC Conf. Comput. Commun. Secure. CCS, 2017, pp. 135–147.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, **“**Explaining and harnessing adversarial examples,” 2014, https://arxiv.org/abs/1412.6572.

[14] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, ‘‘Detecting backdoor attacks on deep neural networks by activation clustering, ’’in Proc. AAAI Workshop Artif. Intell. Saf., Jan. 2019, pp. 66–73.

[15] K. Liu, B. Dolan-Gavitt, and S. Garg, ‘’Fine- pruning: Defending against backdooring attacks on deep neural networks, ’’in Proc. 21st Int. Symp. Res. Attacks Intrusions Def., Sep. 2018, pp. 273–294.

[16] T. Lee, B. Edwards, I. Molloy, and D. Su, ‘‘Defending against neural network model stealing attacks using deceptive perturbations, ’’in Proc. IEEE Secur. Privacy Workshops (SPW), May 2019, pp. 43–49.

[17] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, ‘‘PRADA: Protecting against DNN model stealing attacks, ’’in Proc. IEEE Eur. Symp. Secure. Privacy (EuroS&P), Jun. 2019, pp. 512–527.

[18] S. Chen, N. Carlini, and D. Wagner, ‘’Stateful detection of black-box adversarial attacks, ’’2019, arXiv:1907.05587. [Online]. Available: http://arxiv.org/abs/1907.05587

[19] M. Abadi, A. Chu, I. J. Goodfellow, H. B. Mcmahan, I. Mironov, K. Talwar, and L. Zhang, ‘’Deep learning with differential privacy, ’’in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. CCS, Oct. 2016, pp. 308–318.

[20] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, ‘‘Privacy-preserving deep learning via additively homomorphic encryption’’ IEEE Trans. Inf. Forensics Security, vol. 13, no. 5, pp. 1333–1345, May 2018

[21] B. Jayaraman and D. Evans, ‘’Evaluating differentially private machine learning in practice, ’’in Proc. 28th USENIX Secur. Symp., 2019, pp. 1895–1912.[22] Wiederhold, Gio & McCarthy, John. Arthur Samuel: Pioneer in Machine Learning. IBM Journal of Research and Development. 36. 329 - 331. 10.1147/rd.363.0329, 1992.

[22] Wiederhold, Gio & McCarthy, John. Arthur Samuel: Pioneer in Machine Learning. IBM Journal of Research and Development. 36. 329 - 331. 10.1147/rd.363.0329, 1992.