11-2019

Data security issues in deep learning: attacks, countermeasures, and opportunities

Guowen XU
University of Electronic Science and Technology of China

Hongwei LI
University of Electronic Science and Technology of China

Hao REN
University of Electronic Science and Technology of China

Kan YANG
University of Memphis

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Data Security Issues in Deep Learning: Attacks, Countermeasures, and Opportunities

Guowen Xu, Hongwei Li , Hao Ren, Kan Yang, and Robert H. Deng

Guowen Xu and Hao Ren are with the University of Electronic Science and Technology of China (UESTC); Hongwei Li (corresponding author) is with the University of Electronic Science and Technology of China (UESTC), and also with Peng Cheng Laboratory; Kan Yang is with the University of Memphis; Robert H. Deng is with Singapore Management University.

Published in IEEE Communications Magazine, 2019 November, 57 (12), 116-122.

DOI: 10.1109/MCOM.001.1900091

Abstract:

Benefiting from the advancement of algorithms in massive data and powerful computing resources, deep learning has been explored in a wide variety of fields and produced unparalleled performance results. It plays a vital role in daily applications and is also subtly changing the rules, habits, and behaviors of society. However, inevitably, data-based learning strategies are bound to cause potential security and privacy threats, and arouse public as well as government concerns about its promotion to the real world. In this article, we mainly focus on data security issues in deep learning. We first investigate the potential threats of deep learning in this area, and then present the latest countermeasures based on various underlying technologies, where the challenges and research opportunities on offense and defense are also discussed. Then, we propose SecureNet, the first verifiable and privacy-preserving prediction protocol to protect model integrity and user privacy in DNNs. It can significantly resist various security and privacy threats during the prediction process. We simulate SecureNet under a real dataset, and the experimental results show the superior performance of SecureNet for detecting various integrity attacks against DNN models.

Introduction

Similar to biology, deep learning attempts to imitate humans to think, analyze, and make decisions through continuous training based on a complex topology between neurons. Benefiting from the rich computing resources, deep learning has shown remarkable performance in autonomous driving [1], wireless communication systems [2] and other daily activities of society [3]. For example, by using the massive datasets generated in the wireless network environment, deep learning can be exploited to solve many problems in wireless communication systems, such as decision making [2], resource optimization, and network management [4]. Google has also embedded a deep-learning-based vehicle identification algorithm in autonomous driving. Its onboard laser detection device can accurately identify obstacles such as pedestrians, trains, and vehicles, and calculate the buffer distance and the best route in real time. Undoubtedly, the enormous potential of deep learning has fueled its extensive research, and made it one of the hottest topics in both academia and industry.

To facilitate automated deep learning, many well-known companies (e.g., Google, Microsoft, and Amazon) provide cloud-assisted machine learning services, usually called machine learning as a service (MLaaS). MLaaS provides a range of customized training and prediction services that only require users to upload local data. However, outsourced deep learning also brings about various privacy and security concerns. As mentioned above, while configuring deep learning and enjoying

feedback more conveniently, outsourcing data to untrusted third parties (e.g., a cloud server) puts users' private data at risk. Intuitively, once data is contributed or outsourced to a third party, the data owner will no longer be able to control its utilization as a deposed person. On the other hand, various attacks have been constructed to pose a long-term danger to deep-learning-based frameworks. For example, in a deep neural networks (DNNs) trojan attack [5], the adversary can subtly modify the target model to induce the model to erroneously classify predefined inputs. Such attacks have occurred in public transportation places with dense traffic, such as airports, train stations, and terminals [1, 6], where the adversaries can obtain illegal entry by launching DNN trojan attacks on the face system placed at the inspection port.

Recently, for the weaknesses of deep learning, some results [5, 6] have demonstrated that customized attack against the target model can efficiently undermine the data integrity and availability. On addressing these issues, several defense mechanisms have also been designed for different scenarios. However, the research in this area is still in its infancy. To alleviate these problems, in this article, we first investigate recent works on data security associated with deep learning. Then, to protect the data integrity and privacy in outsourced prediction services, we propose SecureNet, the first verifiable and privacy-preserving protocol to protect model integrity and user privacy in DNNs. Compared to existing models, SecureNet is generalized and can be used for all types of neural network structures without any additional hyper-parameter assumptions. Moreover, SecureNet is also the first practical solution that can guarantee the confidentiality of all user-related private data during the prediction process, while supporting verification of the integrity of a model's parameters outsourced to an untrusted server. Extensive experiments also show the superior performance of SecureNet for detecting various integrity attacks against DNN models.

Deep Learning

Clearly, DNNs are the underlying structure behind the success of deep learning. As described in Fig. 1, traditional DNNs generally consist of an input layer, one or more hidden layers, and an output layer, where each circle represents a neuron associated with an activation function. Traditional activation functions have sigmoid, ReLU, and softmax. Typically, with the input data, the neural network will output the predicted value after applying linear transformations and nonlinear activation functions repeatedly in one or more hidden layers (i.e., feedforward stage). Then, given the real label, the task of DNNs is to minimize the value of the loss function between the real label and its corresponding predicted result for optimal parameter configuration (this process is called Backpropagation). The Stochastic Gradient Descent (SGD) algorithm is usually used to find optimal parameters due to its relative efficiency compared to other methods. As illustrated in Fig. 1, the DNN is trained by iteratively executing feedforward and backpropagation until the accuracy of the model's output meets the predefined accuracy range. After that, the DNN is used to provide predictive services, at which point the user inputs data and only explores feedforward to get the predicted results.

Based on the type of training sample, deep learning can be divided into supervised learning and unsupervised learning. In general, supervised learning uses labeled samples (associated with output values) to achieve tasks such as classification and regression based on those existing output values. Conversely, training samples adopted in unsupervised learning are not labeled, and the target of this type of learning is always to find the structure of data (e.g., clustering). In this article, we focus on supervised learning under DNNs. On the other hand, deep learning can be also divided into two types by two types of training.

Centralized Training: The server interacts with the users to get their local data (training samples). Then it builds an initialized global model in the cloud and iteratively updates the model to obtain the optimal parameters.

Collaborative Training: Collaborative training is also known as distributed, federated, or decentralized training. Each user has a unified local model that has been agreed upon in advance. Then each user optimizes the local model by exchanging parameters with the server frequently.

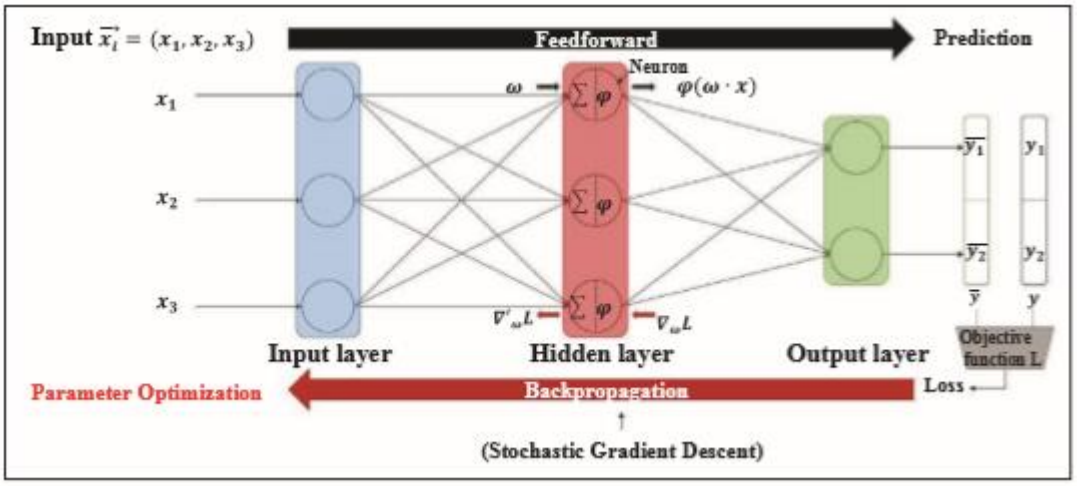

Figure 1: General DNN training process

Threats

In this section, we investigate the data security attacks to deep-learning-based systems (listed in Table 1). Concretely, data security attacks are committed to destroying the data integrity or availability, so as to undermine the training process or make the model output abnormal results. For example, studies have shown that adversaries can fool the autopilot system by interfering with the sensor [2]. Imagine if someone can tamper with the autopilot model to some extent with the sample, which can lead to passenger death.

Attack	Ref	Method	Adversary ability	Attack scenario	Goal
Poisoning	[5]	Solve the "bilevel" optimization regression	White/black box	Collaborataive training	Destroy the availability in linear regression
	[7]	Inject fake data in classification			Lead the model to misclassify the input
	[8]	Inject fake dat in classification			Subvert the training process in injecting
Evasion	[6]	Generate perturbation to perturb objects	White box	Prediction process	Achieve a high targeted misclassification
	[9]	Generate selective audio aversarial examples	Black box		Break speech recognition system
	[10]	Quantify similarity between real/fake data	White box		Break the defense technology based on distillation

Table 1: Attacks against data security in deep learning

As with the workflow phase, the attack scenario is subject to change depending on the prior knowledge acquired by the attacker on the target model. If the adversary model has full access rights to all information of a model, such as model architecture, parameter details, and training data, it will show a high attack success rate, but is rare in reality. Conversely, if the adversary has limited access rights, such as only to the model's predictive interface, it is hard to attack and may require alternative

methods such as substitute model or data. Therefore, based on the access authority of adversaries to the DNN model, two types of attackers are considered in our article, that is, black-box attacker and white-box attacker.

Black-Box Attacker: A black-box attacker can make prediction queries on the DNN model, but it is prohibited from accessing the model's internal information such as parameter configuration, optimization procedures, and training sets.

White-Box Attacker: A white-box attacker, in addition to having access to the query function, can acquire much more information including the DNN model description, architecture, and some training samples. Obviously, white-box attackers are more powerful than black-box attackers. However, due to the inaccessibility of target model information, the latter tends to be a more realistic assumption.

In terms of attacks against data security, such as data integrity and availability, we mainly discuss two major attacks at different stages of deep learning: poisoning attack and evasion attack. In the following subsections, we describe the definition and research progress in these two types of attacks in detail.

Poisoning Attack

The goal of the poisoning attack is to destroy the availability of the output during the training process, thus making DNNs perform poorly in the subsequent prediction service. Its main method is to mislead the network to make incorrect predictions by carefully crafting poisoning samples (also called adversarial examples). Following the logic flow ahead, we also analyze the effects of the poisoning attack in the cases of the black-box attacker and white-box attacker, respectively.

Black-Box Attacker: In the poisoning attack, the black-box adversary is only allowed to inject a small portion of samples into the training process without the authority to access the model and training process. In general, a good poisoning attack satisfies three constraints in the case of black-box accession:

• There is nothing to know about the model.

• Only a small amount of custom training data is allowed to be injected.

• Poisoning samples are hard for humans to detect.

For instance, Jagielski et al. [5] first considered investigating attacks against linear regression under black-box access to the training model. By optimizing traditional classification-based attack algorithms, they proposed a theory-based optimization framework for adjusting regression models, and designed a fast attack algorithm only requiring limited knowledge. Suciu et al. [7] proposed StingRay, a targeted poisoning attack that overcame the limitations of prior attacks. StingRay systematically outlined the minimum prior knowledge required by the adversary under different attack goals, and gave a very effective black-box attack even if the target model is protected by existing defense mechanisms.

White-Box Attacker: In a poisoning attack, a white-box attacker has powerful access rights to the model's parameters, architecture, and training details, and can use this information to carefully construct poisoning samples. For example, Yuan et al. [8] presented a perfect-knowledge (PK) attack under various scenarios. Although the PK attack scenario is an unrealistic setting, the results show that it is nearly five times more accurate than a normal black-box attack. Suciu et al. [7] also proposed a poisoning attack with black-box access to the target model, which is effective against four real-world classification tasks, and can bypass the detection of two existing mainstream anti-poisoning defenses. Jagielski et al. [5] designed a poisoning attack against linear regression models. They first

used an existing poisoning attack as a basic regression attack model. After that, an optimization framework for poisoning attacks against regression tasks was designed, in which the objective function, initialization strategy, and optimization variables can be selected to maximize the impact of attacks on targeted models and datasets.

Evasion Attack

An evasion attack is often used during the prediction process, which misleads the DNN model into predicting an incorrect label by carefully adding noises to the real samples. From a geometric point of view, the purpose of an evasion attack is to move a real sample from one class to another by destroying the integrity of the original sample. Similarly, we also analyze the effects of the evasion attack in the cases of black-box attacker and white-box attacker, respectively.

Black-Box Attacker: In evasion attack, the black-box attacker has no information about the training samples and the targeted model. The information available is only the format of the training data and the output of the model. In the real world, it is not easy to access an already trained model or training set. Although there is a large amount of public data (images, sounds, videos, etc.), the internal data used to train the industrial model is still confidential. Also, an attacker cannot access models contained in the mobile device. Therefore, the ability of a blackbox attacker is in line with reality. For example, Kwon et al. [9] proposed an evasion attack on a speech recognition system that generates selective audio adversarial examples to move a real sample from one class to another. Experiments also show that their selective audio adversarial examples can achieve an attack success rate of at least 91.67.

White-Box Attacker: In evasion attack, the white-box attacker, in addition to having access to the predictive interface, can acquire much more information including the DNN model description, architecture, and some training samples. For example, Kevin Eykholt et al. [6] have shown that the most advanced DNNs are vulnerable to such attacks even if small noise perturbations are added to the input. They further proposed Robust Physical Perturbations (RP2), a general attack algorithm to create robust visual adversarial perturbations under different physical conditions. Testing results demonstrated the high targeted misclassification rate in the road signal classification scenarios. Similarly, in view of the existing defense methods using defensive distillation technology, Carlini et al. [10] designed three new attack algorithms to prove that defensive distillation does not significantly increase the robustness of neural networks.

Challenges and Opportunities: White-box attacks appear to be very effective, and often bypass most defense mechanisms. However, since adversaries need to access the DNN model description, architecture, and training samples, it is not easy to implement a white-box attack in reality. For example, in collaborative training, each user only shares its local parameters to the server without revealing its local dataset. In this case, it is difficult for the adversary to obtain the user's training samples by only analyzing the shared gradients. Black-box attacks consider weaker adversary models. However, to obtain more information, an effective black-box attack has to interact with the target model multiple times, which inevitably increases the communication overheads. More seriously, multiple interactions with the target in violation of common sense will also put the adversary's identity at high risk. Therefore, this is a direction worth studying to design an effective attack that fits the realworld scenario.

Countermeasures

There are many approaches proposed for resisting various threats of deep learning. In this section, we also discuss the existing defense methods for poisoning and evasion attacks, respectively (listed in Table 2).

Countermasures	Ref	Method	Threat model	Application scenario	Design goal
Against poisoning attack	[11]	Utilize influence functions to trace the training set	Active	Collaborative training	Eliminate false data injected by adversaries
	[12]	Defense strategy called sphere defense and slab	Active		
	[13]	Utilize key sharing protocol-based technologies	Active		Protect integrity of calculation results.
Against evasion attack	[14]	Explore specialized interactive proof protocol	Semi-honest	Prediction process	Verify the correctness of results from the server
	[15]	Exploit sensitive-sammples-based technology	Active		Protect integrity of parameters in the DNNs

Table 2: Protection approaches for data security in deep learning

Defense against Poisoning Attack

In general, one of the main ways to defend against poisoning attacks is to design efficient detection mechanisms, which can rapidly detect abnormal samples and eliminate these poisoned data during training. For example, Koh et al. [11] first used influence functions to trace and explain the correlation between prediction and training sets. They demonstrated that the influence functions can be widely used for malicious data detection in poisoning attacks even in nonconvex and non-differentiable models. Steinhardt et al. [12] proposed a defense scheme by constructing approximate upper bounds on the loss across a broad family of attacks. Further, they designed two efficient defense strategies called sphere defense and slab defense to remove outliers (i.e., data suspected of being injected by the adversary) that are outside the applicable set. In this way, the false data in the DNN model can be effectively detected and filtered. Xu et al. proposed VerifyNet [13]. It uses key sharing protocols to protect the integrity of training samples, thereby preventing malicious adversaries from tampering with training samples and calculation results.

Defense against Evasion Attack

In terms of evasion attack, the current mainstream direction is to design an approach effectively verifying the integrity of the target model. For example, Ghodsi et al. [14] proposed SafetyNets, a data integrity verification framework that can detect whether a malicious cloud server implements an evasion attack by modifying the pre-agreed protocol. Specifically, to achieve end-to-end verifiability, they used a specialized interactive proof (IP) protocol to verify the results of activation layers, and Thaler's IP protocol [14] for matrix multiplication. He et al. [15] exploited Sensitive-Samples as fingerprints of DNN models to check the integrity of results returned from the server, even with black-box access rights. They claimed that the Sensitive-Samples-based methodology is very sensitive to changes in parameters, where even small misrepresentations are difficult to escape.

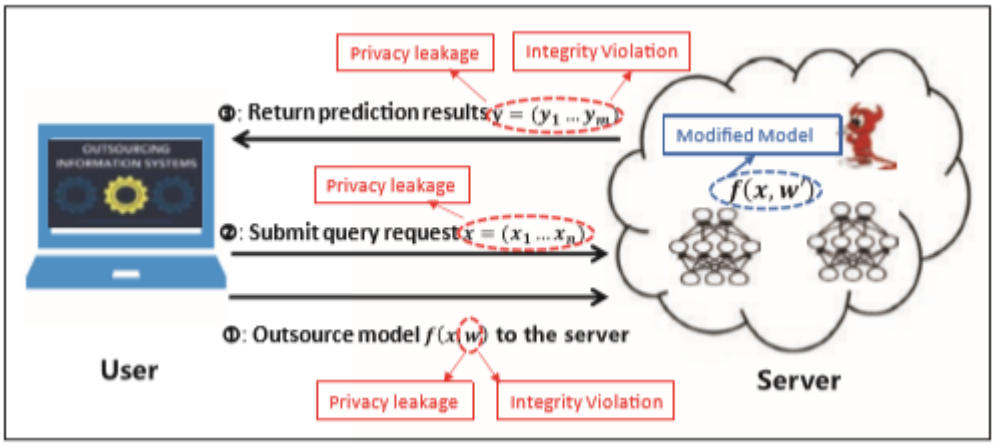

Figure 2: System model

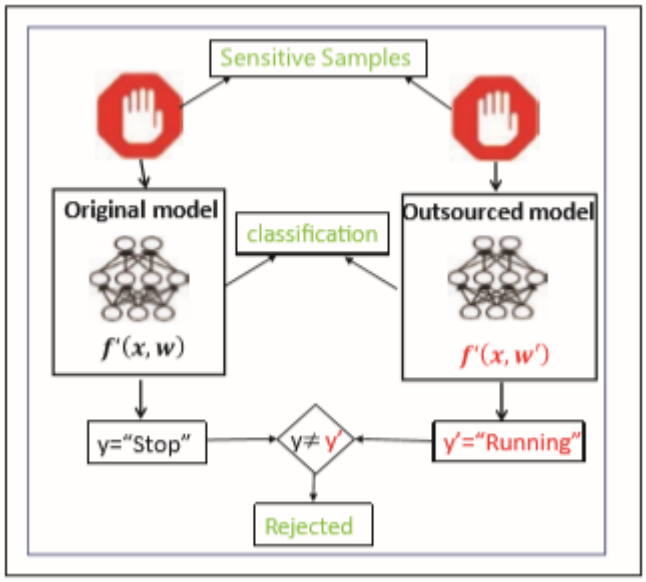

Figure 3. Example of verifying the model integrity through sensitive samples

Challenges and Opportunities: Several secure approaches [12, 14] have been successively proposed to alleviate the effects of poisoning and evasion attacks. However, these schemes either support few activation functions or require unrealistic prior knowledge. Also, due to the flexibility of the attack, it is difficult for existing defense methods to effectively detect highly concealed attacks, such as target poisoning and evasion attacks. Moreover, no solution has been proposed to support both verifiability and data privacy protection in the outsourced prediction process. Specifically, in outsourced prediction services, once a user outsources its model to the server, it may return incorrect results to users by tampering with the model's parameters. Besides, it is possible that the server abuses the user's model parameters, and even exploits the inference service to collect the user's sensitive data. While some generalized techniques such as arithmetic circuits can be applied to protect the model's integrity during an outsourced prediction process, the huge overhead in the implementation process leaves them stuck in the theoretical stage. Recently, trusted hardware such as SGX and TrustZone are also being adopted to provide trusted execution environments (TEEs) for deep learning. However, in addition to increasing operating costs, trusted hardware alone does not achieve the dual goals of data privacy and model integrity. Hence, it is worthwhile to delve into designing a lightweight learning framework that can protect both data security and privacy in DNNs.

Example Scheme to Deal with some of the Proposed Challenges

As discussed above, there are many attacks in deep learning. In this section, we propose SecureNet as an example that only focuses on the verifiability and data privacy protection in the outsourced prediction process. It is worth emphasizing that our SecureNet is the first verifiable and privacy-preserving prediction protocol to protect the model's integrity and user's privacy in DNNs. To present our model details in an easy-to-understand way, in the following sections we first describe the system model and designed goals considered in our protocol. Then we describe the technical details of our SecureNet.

System Architecture and Designed Goal

As shown in Fig. 2, our SecureNet consists of two generic entities, a user and a cloud server. To achieve automated outsourcing prediction services, the user first outsources its DNN model to the server. As described in Fig. 2, the w denotes the parameters of DNNs, and x represents the variable of input. Please note that we do not make any assumptions about the DNN training process, which means that the user can obtain the DNN model by training locally or get it from open source resources. After receiving the DNN model from the user, the server allocates resources for this model and releases the application programming interface (API) for prediction. In the end, the user submits the query request (e.g., classification and regression task) to the server and receives the predicted results.

However, the outsourced prediction service also brings about several security and privacy issues (shown in the red font in Fig. 2). Specifically, once a user outsources its DNN model to the server, a malicious adversary may intentionally tamper with the model's parameters for obtaining certain benefits. For example, face recognition has been widely used in monitoring systems in large public places. To bypass the recognizer or make it malfunction, the adversary would seek to modify the model parameters without being noticed. On the other hand, once the user uploads the local model to the server, it is possible that the server abuses the user's model parameters and even exploits the prediction service to collect the user's sensitive data, such as query requests and predicted results.

Therefore, the goal of our SecureNet is to provide secure and privacy-preserving prediction service. To achieve this, we first transform the complex nonlinear activation functions (e.g., Sigmoid and ReLU) of DNNs into polynomials. Then leveled homomorphic encryption (LHE) is used to encrypt all user-related private data. In the end, we generate generic Sensitive-Samples to verify the integrity of parameters outsourced to the server. In the following sections, we describe these technologies used in our SecureNet in detail.

Privacy-preserving Prediction based on LHE

We know that homomorphic encryption, SMC, and differential privacy are three main underlying structures exploited to protect data privacy in deep learning. However, in the outsourced prediction process, since we only require the server to execute the prediction program of DNNs without sharing secrets, SMC is not suitable for our application scenarios. Similarly, the statistical operation is not involved in the prediction process; hence, the use of differential privacy technology will inevitably lead to errors that are difficult to offset. Fully homomorphic encryption (FHE) is a potential solution. However, as discussed before, it leads to huge computational overheads. To address the above problems, we exploit LHE in our SecureNet to encrypt users' private data. LHE is more efficient than FHE, but only supports limited addition and multiplication in the ciphertext. This shortcoming makes it impossible for LHE to calculate complex activation functions such as ReLU, Sigmoid, and Tanh

under ciphertext. To combat that, we adopt a function approximation algorithm to convert nonlinear activations to polynomials. Based on the Weierstrass approximation theorem, we can prove that most of the activation functions existing in DNNs can be approximated by polynomials. Hence, in SecureNet, the original DNN model first will be transformed into another model that only contains addition and multiplication operations. Then the transformed model will be encrypted by LHE and sent to the cloud. In the end, the user submits an encrypted query request (encrypted by LHE) to the server and receives the ciphertext results.

Verify Model Integrity through Sensitive-Samples

We have adopted LHE to protect the privacy of user data such as the model's parameters, users' query requests, and predicted results in the outsourcing prediction process. However, Fig. 2 shows that the adversary can still change the original model to a targeted model by tampering with the parameter O, thus to launch data integrity attacks. To address this problem, our main idea is to generate a small set of test samples (denoted as Sensitive-Samples) to check whether the adversary has changed the original model. Specifically, by solving the optimization problem, we first generate a set of Sensitive-Samples that are very sensitive to model parameter changes. Then, as shown in Fig. 3, we submit these Sensitive-Samples to the cloud and obtain the corresponding outputs (e.g., classification results). By comparing the outputs under the outsourced model with the outputs of the original model, the user can verify whether the outsourced model is intact or modified.

Summary: Based on the above description, we claim that our SecureNet can provide secure and privacy-preserving prediction service. Concretely, we first transform the complex nonlinear activation functions of DNNs into polynomials. Then we adopt LHE to support privacy-preserving prediction. In the end, we generate generic Sensitive-Samples to verify the integrity of parameters outsourced to the server.

Performance Evaluation

We perform our simulation experiments on the MNIST database (http://yann.lecun.com/exdb/ mnist/), which holds a test set of 10,000 examples and a training set of 60,000 examples. For simplicity, all experiments were done under a convolutional neural network (CNN), which has two fully connected layers, one pooling layer, and two convolutional layers.

In theory, our Sensitive-Samples are generic and resistant to diverse data integrity attacks. To evaluate the detection accuracy, in our experiments, we consider four common integrity attacks in the outsourcing prediction process.

Neural Network Trojan Attack (NNTA [6]): The adversary can replace the selected parameters with Trojans. This allows the outsourced model to return the correct results for normal inputs, while returning the results desired by the adversary for inputs containing Trojans.

Targeted Poisoning Attack (TPA [5]): To make DNNs misclassify the inputs to targeted outputs, a malicious server can retrain outsourced models using carefully prepared training samples.

Model Compression Attack (MCA): To reduce the cost, a malicious server can compress the original model to a simple model without significantly changing its prediction accuracy.

Arbitrary Weights Modification (AWM): This is very understandable: an adversary (e.g., in the cloud) can change any parameter of the outsourced model.

As shown in Table 3, we consider the case of a server returning Top-k (k = 1, 3, 5) classification labels to users. Since we only have the black-box way to access the outsourced model, the less information included in outputs (from Top-5 (most) to Top-1 (least)), the harder it is to detect tampering using Sensitive-Samples. Our sensitive samples can detect model tampering with high accuracy (> 87.3 percent), even returning Top-1 result, and the model changes can be completely detected if the server returns the Top-3 ciphertext results to the user. Hence, based on the results of Table 3, we demonstrate that our SecureNet is resistant to diverse data integrity attacks.

Type of attack	No. of samples	Top-1	Top-3	Top-5
NNTA	1	94%	100%	100%
	3	98.4%	100%	100%
	5	100%	100%	100%
TPA	1	87.30%	100%	100%
	3	99.30%	100%	100%
	5	99.30%	100%	100%
MCA	1	80.29%	100%	100%
	3	94.48%	100%	100%
	5	99.17%	100%	100%
AWM	1	97.23%	100%	100%
	3	99.78%	100%	100%
	5	100%	100%	100%

Table 3: Detection accuracy of our SecureNet with different attacks

Conclusion

In this article, we have investigated the research status on data security issues in deep learning and explored some of the challenges that need to be addressed. On the basis of previous studies, we have also extracted several future research opportunities that deserve to be explored in depth. In the end, we propose SecureNet as an example to protect the model's integrity and the user's privacy in DNNs. As the core of our future work, we are committed to improving the efficiency of SecureNet, including reducing the communication and computation overhead of the detection and encryption process.

Acknowledgement

This work is supported by the National Key R&D Program of China under Grants 2017YFB0802300 and 2017YFB0802000, the National Natural Science Foundation of China under Grants 61972454, 61802051, 61772121, 61728102, and 61472065, the Peng Cheng Laboratory Project of Guangdong Province PCL2018KP004, the Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS201804.

References

[1] G. Xu et al., “Efficient and Privacy-Preserving Truth Discovery in Mobile Crowd Sensing Systems,” IEEE Trans. Vehic. Tech., vol. 68, no. 4, Jan. 2019, pp. 3854–65.

[2] S. Otoum, B. Kantarci, and H. T. Mouftah, “On the Feasibility of Deep Learning in Sensor Network Intrusion Detection,” IEEE Networking Letters, vol. 1, no. 2, Feb. 2019, pp. 68–71.

[3] M. Usman et al., “P2DCA: A Privacy-Preserving-Based Data Collection and Analysis Framework for IOMT Applications,” IEEE JSAC, vol. 37, no. 6, June 2019, pp. 1222–30.

[4] M. Aloqaily et al., “Data and Service Management in Densely Crowded Environments: Challenges, Opportunities, and Recent Developments,” IEEE Commun. Mag., vol. 57, no. 4, Apr. 2019, pp. 81–87.

[5] M. Jagielski et al., “Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning,” Proc. IEEE Security & Privacy, May 2018, pp. 19–35.

[6] K. Eykholt et al., “Robust Physical-World Attacks on Deep Learning Visual Classification,” Proc. IEEE CVPR, June 2018, pp. 1625–34.

[7] O. Suciu et al., “When Does Machine Learning Fail? Generalized Transferability for Evasion and Poisoning Attacks,” Proc. USENIX Security, Aug. 2018, pp. 1299–1316.

[8] X. Yuan et al., “Adversarial Examples: Attacks and Defenses for Deep Learning,” IEEE Trans. Neural Networks Learning Systems, vol. 30, no. 9, Jan. 2019, pp. 2805–42.

[9] H. Kwon et al., “Selective Audio Adversarial Example Evasion Attack on Speech Recognition System,” IEEE Trans. Info. Forensics and Security, vol. 15, June 2019, pp. 525–38.

[10] N. Carlini and D. Wagner, “Towards Evaluating the Robustness of Neural Networks,” Proc. IEEE Security & Privacy, May 2017, pp. 39–57.

[11] P. W. Koh and P. Liang, “Understanding Black-Box Predictions via Influence Functions,” Proc. ACM ICML, Aug., 2017, pp. 1–11.

[12] J. Steinhardt, P. W. W. Koh, and P. S. Liang, “Certified Defenses for Data Poisoning Attacks,” Advances in Neural Info. Processing Systems, Dec. 2017, pp. 3517–29.

[13] G. Xu et al., “Verifynet: Secure and Verifiable Federated Learning,” IEEE Trans. Info. Forensics and Security, vol. 15, July 2019, pp. 911–26.

[14] Z. Ghodsi, T. Gu, and S. Garg, “Safetynets: Verifiable Execution of Deep Neural Networks on an Untrusted Cloud,” Advances in Neural Info. Processing Systems, Dec. 2017, pp. 4672–81.

[15] Z. He, T. Zhang, and R. B. Lee, “Verideep: Verifying Integrity of Deep Neural Networks through Sensitive-Sample Fingerprinting,” Proc. IEEE CVPR, June, 2018.

Biographies

Guowen Xu is currently a Ph.D. student at the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include network security and verifiable deep learning.

Hongwei Li is currently a professor at the School of Computer Science and Engineering, UESTC, and also with the Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China. His research interests include network security and applied cryptography.

Hao Ren is currently a Ph.D. student at the School of Computer Science and Engineering, UESTC. His research interests include cryptography and security and privacy of outsourcing data.

Kan Yang is currently an assistant professor with the Department of Computer Science, University of Memphis. His research interests include security and privacy in big data and distributed systems.

Robert H. Deng is currently the AXA Chair Professor of Cybersecurity, Singapore Management University. His research interests are in the areas of data security and privacy, cloud security, and IoT security.