www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245

A NOVEL APPROACH TO DATA POISONING DETECTION IN DISTRIBUTED MACHINE LEARNING NETWORKS

[1]D. Saikrishna,[2]G.Ajay
[1]Assistant Professor, [2]MCA Student
Department Of MCA Student
Sree Chaitanya College of Engineering, Karimnagar

ABSTRACT:

Massive dataset training is possible with distributed machine learning (DML) when no single node can produce correct results in a reasonable amount of time. However, compared to the non-distributed system, this will unavoidably expose more possible targets to attackers. We divide DML into basic-DML and semi-DML in this study. The centre server assigns learning assignments to dispersed machines in basic-DML and compiles the learning outcomes. In addition to its responsibilities in basic-DML, the centre server in semi-DML also invests resources in dataset learning. To identify the contaminated data, we first propose a new data poison detection approach for basic-DML that makes use of a cross-learning process. A mathematical model is developed to determine the ideal number of training loops after we demonstrate that the suggested cross-learning process will produce training loops. Then, with the use of the central resource, we provide an enhanced data poison detection technique for semi-DML to increase learning protection. An optimum resource allocation strategy is created in order to make effective use of the system's resources. According on simulation data, in the basic-DML scenario, the suggested strategy can greatly increase the final model's accuracy by up to 20% for support vector machines and 60% for logistic regression. Additionally, the enhanced data poison detection system with optimum resource allocation may reduce resource waste by 20–100% in the semi-DML situation.

1. INTRODUCTION

Distributed machine learning (DML) has been widely used in distributed systems [1], [2], where no single node can get the intelligent decision from a massive dataset within an acceptable time [3]–[6]. In a typical DML system [7], a central server has a tremendous amount of data at its disposal. It divides the dataset into different parts and disseminates them to distributed workers who perform the training tasks and return their results to the center [8]–[10]. Finally, the center integrates these results and outputs the eventual model.

Unfortunately, with the number of distributed workers increasing, it is hard to guarantee the security of each worker. This

Index in Cosmos
Nov 2024, Volume 14, ISSUE 4
UGC Approved Journal

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245

lack of security will increase the danger that attackers poison the dataset and manipulate the training result. Poisoning attack [11]–[13] is a typical way to tamper the training data in machine learning. Especially in scenarios that newly generated datasets should be periodically sent to the distributed workers for updating the decision model, the attacker will have more chances to poison the datasets, leading to a more severe threat in DML.

Such vulnerability in machine learning has attracted much attention from researchers. Dalvi et al. [14] initially demonstrated that attackers could manipulate the data to defeat the data miner if they have complete information. Then Lowdet al. [15] claimed that the perfect information assumption is unrealistic, and proved the attackers can construct attacks with part of the information. Afterwards, a series of works were conducted [16]–[23], focusing on non-distributed ma-chine learning context. Recently, there are a couple of efforts devoted in preventing data from being manipulated in DML.F or example, Zhang et al. [24] and Esposito et al. [25] used game theory to design a secure algorithm for distributed support vector machine (DSVM) and collaborative deep learning, respectively. However, these schemes are designed for specific DML algorithm and cannot be used in general DML situations. Since the adversarial attack can mislead various machine learning algorithms, a widely applicable DML protection mechanism is urgent to be studied various machine learning algorithms, a widely applicable DML protection mechanism is urgent to be studied.

In this paper, we classify DML into basic distributed machine learning (basic-DML) and semi distributed ma-chine learning (semi-DML), depending on whether the center shares resources in the dataset training tasks. Then, we present data poison detection schemes for basic-DML andsemi-DML respectively. The experimental results validate the effect of our proposed schemes. We summary the main contributions of this paper as follows.

We put forward a data poison detection scheme for basic-DML, based on a so-called cross-learning data assignment mechanism. We prove that the cross-learning mechanism would consequently generate training loops, and provide a mathematical model to find the optimal number of training loops which has the highest security.

We present a practical method to identify abnormal training results, which can be used to find out the poisoned datasets at a reasonable cost.

For semi-DML, we propose an improved data poison detection scheme, which can provide better learning protection. To efficiently utilize the system resources, an optimal resource allocation scheme is developed.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245

The rest of this paper is organized as follows. We firstly introduce the system model in Section II and the threat model in Section III. Then, the data poison detection scheme in basic-DML and semi-DML are described in detail in Section IV and Section V, respectively. Simulation results demonstrate the effectiveness of proposed schemes in Section VI, which is followed by the summary and future work.

2. LITERATURE SURVEY

G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," IEEE Communications Magazine, vol. 56, no. 8, pp. 48–54, 2018.

Mobile edge computing (MEC) has emerged as a promising paradigm to realize user requirements with low-latency applications. The deep integration of multi-access technologies and MEC can significantly enhance the access capacity between heterogeneous devices and MEC platforms. However, the traditional MEC network architecture cannot be directly applied to the Internet of Vehicles (IoV) due to high speed mobility and inherent characteristics. Furthermore, given a large number of resource-rich vehicles on the road, it is a new opportunity to execute task offloading and data processing onto smart vehicles. To facilitate good merging of the MEC technology in IoV, this article first introduces a vehicular edge multi-access network that treats vehicles as edge computation resources to construct the cooperative and distributed computing architecture. For immersive applications, co-located vehicles have the inherent properties of collecting considerable identical and similar computation tasks. We propose a collaborative task offloading and output transmission mechanism to guarantee low latency as well as the application- level performance. Finally, we take 3D reconstruction as an exemplary scenario to provide insights on the design of the network framework. Numerical results demonstrate that the proposed scheme is able to reduce the perception reaction time while ensuring the application-level driving experiences.

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning." in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), vol. 16. USENIX Association, 2016, pp. 265–283.

TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. TensorFlow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across many

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245

machines in a cluster, and within a machine across multiple computational devices, including multicore CPUs, general-purpose GPUs, and custom designed ASICs known as Tensor Processing Units (TPUs). This architecture gives flexibility to the application developer: whereas in previous "parameter server" designs the management of shared state is built into the system, TensorFlow enables developers to experiment with novel optimizations and training algorithms. TensorFlow supports a variety of applications, with particularly strong support for training and inference on deep neural networks. Several Google services use TensorFlow in production, we have released it as an open-source project, and it has become widely used for machine learning research. In this paper, we describe the TensorFlow dataflow model in contrast to existing systems, and demonstrate the compelling performance that TensorFlow achieves for several real-world applications.

3. EXISTING SYSTEM

Unfortunately, with the number of distributed workers increasing, it is hard to guarantee the security of each worker. This lack of security will increase the danger that attackers poison the dataset and manipulate the training result. Poisoning attack is a typical way to tamper the training data in machine learning. Especially in scenarios that newly generated datasets should be periodically sent to the distributed workers for updating the decision model, the attacker will have more chances to poison the datasets, leading to a more severe threat in DML.

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well

4. PROPOSED SYSTEM

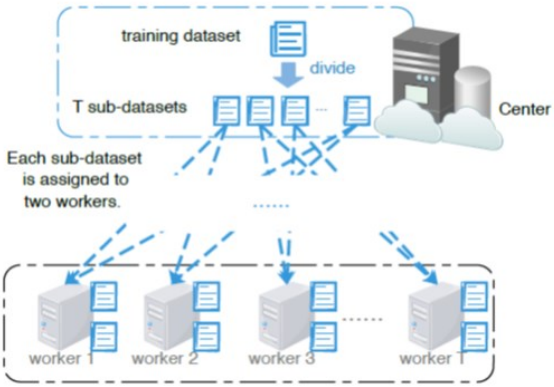
DML into basic distributed machine learning (basic-DML) and semi distributed machine learning (semi-DML), depending on whether the center shares resources in the dataset training tasks. Then, we present data poison detection schemes for basic-DML and semi-DML respectively. The experimental results validate the effect of our proposed schemes.

We classify DML into basic-DML and semi-DML, which are shown in Fig.1, respectively. Both of the two scenarios have a center, which contains a database, a computing server, and a parameter server. However, the center provides different functions in these two scenarios. In the basic-DML scenario, the center has no spare computing resource for sub-dataset training, and will send all the sub-datasets to the distributed workers. Therefore, in the basic-

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245

DML, the center only integrates the training results from distributed workers by the parameter server.

5. SYSTEM ARCHITECTURE



6. MODULES

To implement this project we have designed following modules.

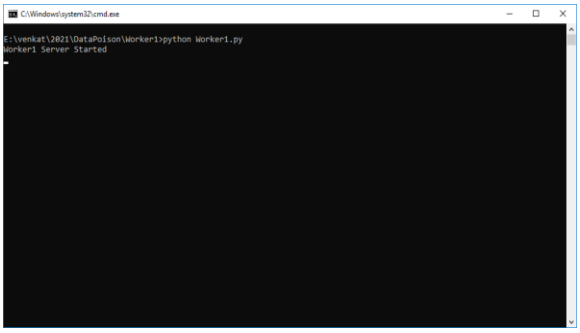
Worker1: This is a worker node which accept divided dataset from center server and then build existing SVM model and Basic DML model and then calculate accuracy of both algorithms and send result back to center server

Worker2: This is another worker node which accept other half of dataset and then run existing SVM and Basic DML and send accuracy back to center server.

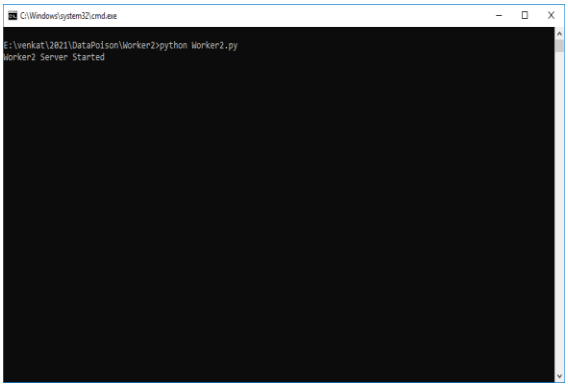
CenterServer: This is a center server which upload dataset to application and then divide dataset into two equal parts and then distribute each part to worker 1 and 2 and then collect result. This server will run semi DML and calculate its accuracy also.

7. SCREEN SHOTS

To run project first double click on 'run.bat' file from Worker1 folder to start worker 1 node and to get below screen



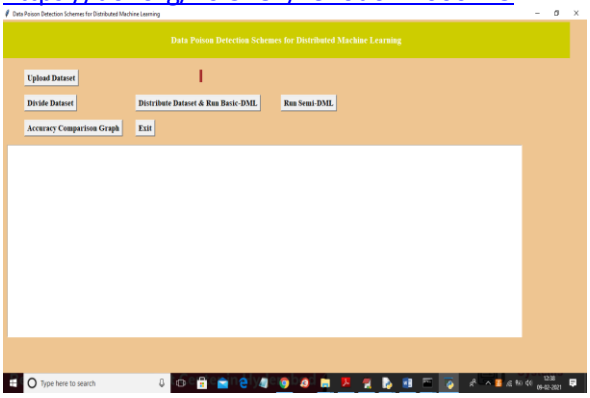
In above screen worker 1 server started and now double click on 'run.bat' file from worker2 folder to start worker 2



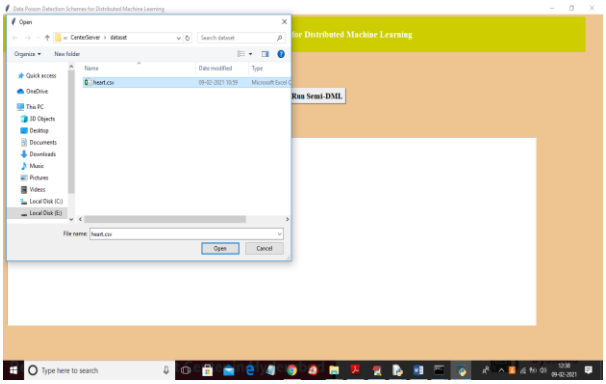
In above screen worker2 server started and now double click on 'run.bat' file from 'CenterServer' folder to start distributed server and to get below screen

Index in Cosmos
Nov 2024, Volume 14, ISSUE 4
UGC Approved Journal

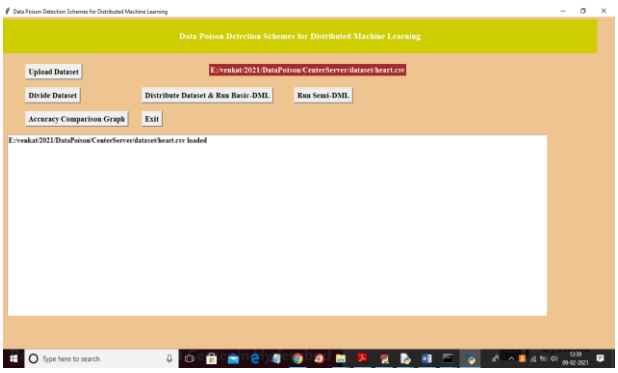
www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245



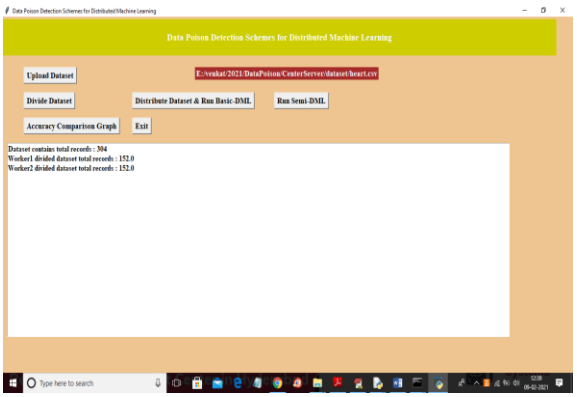
In above screen click on 'Upload Dataset' button to upload dataset and to get below screen



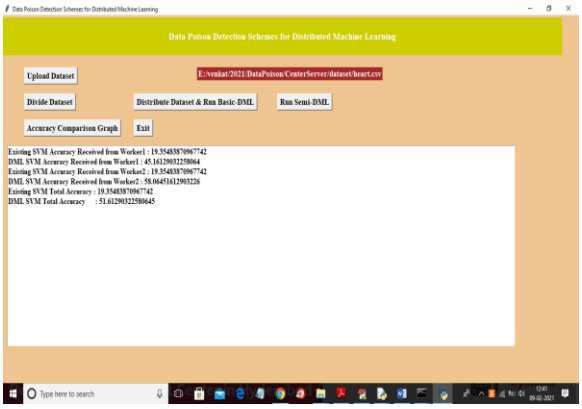
In above screen selecting and uploading 'heart.csv' file and then click on 'Open' button to load dataset and to get below screen



In above screen dataset loaded and now click on 'Divide Dataset' button to divide dataset into 2 equal parts



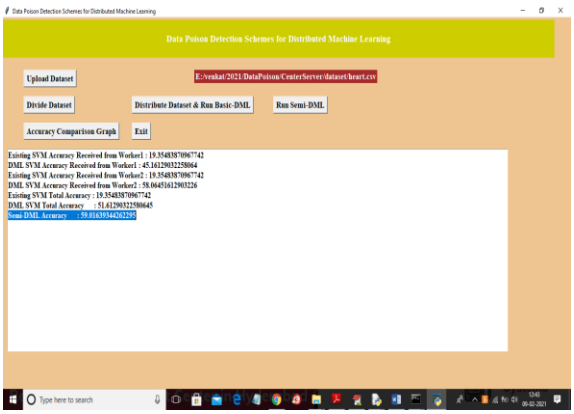
In above screen dataset contains 304 records and equally distributed to 2 parts and now click on 'Distribute Dataset & Run Basic-DML' button to distribute dataset to 2 workers and then get accuracy result



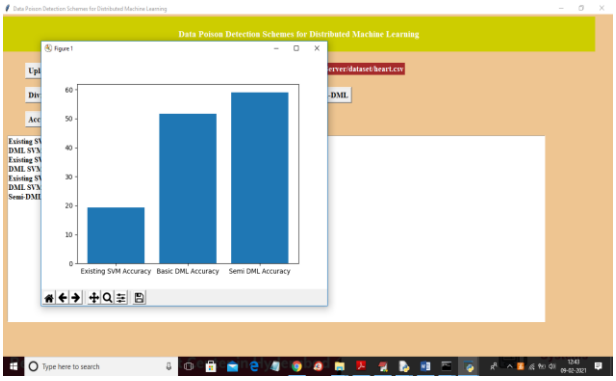
In above screen we got result from 2 worker nodes for existing SVM accuracy and propose DML accuracy and in above screen we can see existing SVM accuracy is 19% when data poison exists in dataset and after removing data poison using DML technique we got 51% accuracy and now click on 'Run

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245

Semi-DML' button to allow center server to devote resources to DML and then remove poison from dataset and then calculate accuracy



In above screen Semi-DML accuracy is 59% and now click on 'Accuracy Comparison Graph' button to get below graph



In above screen x-axis contains algorithm name and y-axis represents accuracy and from above graph we can conclude that Basic-DML and Semi-DML accuracy is better than existing SVM accuracy. In below worker screens also we can see accuracy values

CONCLUSION

In both basic-DML and semi-DML scenarios, we spoke about the data poison detection systems. In the basic-DML scenario, the data poison detection system uses a threshold of parameters to identify the sub-datasets that are contaminated. Additionally, we developed a mathematical model to examine the likelihood of identifying threats with varying training loop counts. Additionally, we demonstrated the best resource allocation in the semi-DML situation as well as an enhanced data poison detection technique. According to simulation data, the suggested strategy can improve model accuracy by up to 20% for support vector machines and 60% for logistic regression in the basic-DML scenario. Compared to the other two techniques without optimum resource allocation, the enhanced data poison detection scheme with optimal resource allocation may reduce resource waste by 20–100% in the semi-DML scenario.

References:

[1] G. Qiao, S. Leng, K. Zhang, and Y. He, ''Collaborative task offloading in vehicular edge multi–access networks,'' IEEE Commun. Mag., vol. 56, no. 8, pp. 48–54, Aug. 2018.

[2] K. Zhang, S. Leng, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, ''Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks,'' IEEE

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245

Internet Things J., vol. 6, no. 2, pp. 1987–1997, Apr. 2019.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, ‘‘Tensorflow: A system for large-scale machine learning,’’ in Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI), vol. 16, 2016, pp. 265–283.

[4] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, ‘‘Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems,’’ Dec. 2015, arXiv:1512.01274. [Online]. Available: https://arxiv.org/abs/1512.01274

[5] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, ‘‘Machine learning on big data: Opportunities and challenges,’’ Neurocomputing, vol. 237, pp. 350–361, May 2017.

[6] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, ‘‘Networking for big data: A survey,’’ IEEE Commun. Surveys Tuts., vol. 19, no. 1, pp. 531–549, 1st Quart., 2016.

[7] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, ‘‘Scaling distributed machine learning with the parameter server,’’ in Proc. 11th USENIX Symp. Operating Syst. Design Implement. (OSDI), vol. 14, 2014, pp. 583–598.

[8] B. Fan, S. Leng, and K. Yang, ‘‘A dynamic bandwidth allocation algorithm in mobile networks with big data of users and networks,’’ IEEE Netw., vol. 30, no. 1, pp. 6–10, Jan. 2016.

[9] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, ‘‘Home M2M networks: Architectures, standards, and QoS improvement,’’ IEEE Commun. Mag., vol. 49, no. 4, pp. 44–52, Apr. 2011.

[10] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, and Y. Zhang, ‘‘Blockchain and deep reinforcement learning empowered intelligent 5G beyond,’’ IEEE Netw., vol. 33, no. 3, pp. 10–17, May/Jun. 2019.

[11] L. Mu noz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, ‘‘Towards poisoning of deep learning algorithms with back-gradient optimization,’’ in Proc. 10th ACM Workshop Artif. Intell. Secur., 2017, pp. 27–38.

[12] S. Yu, G. Wang, X. Liu, and J. Niu, ‘‘Security and privacy in the age of the smart Internet of Things: An overview from a networking perspective,’’ IEEE Commun. Mag., vol. 56, no. 9, pp. 14–18, Sep. 2018.

[13] S. Alfeld, X. Zhu, and P. Barford, ‘‘Data poisoning attacks against autoregressive models,’’ in Proc. 13th AAAI Conf. Artif. Intell., Feb. 2016.

[14] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, ‘‘Adversarial classification,’’

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86
https://doi.org/10.5281/zenodo.14066245
in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2004, pp. 99–108.

[15] D. Lowd and C. Meek, ‘‘Adversarial learning,’’ in Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2005, pp. 641–647.

[16] B. Biggio and F. Roli, ‘‘Wild patterns: Ten years after the rise of adversarial machine learning,’’ Pattern Recognit., vol. 84, pp. 317–331, Dec. 2018.

[17] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, ‘‘A survey on security threats and defensive techniques of machine learning: A data driven view,’’ IEEE Access, vol. 6, pp. 12103–12117, 2018.

[18] Z. Yin, F. Wang, W. Liu, and S. Chawla, ‘‘Sparse feature attacks in adversarial learning,’’ IEEE Trans. Knowl. Data Eng., vol. 30, no. 6, pp. 1164–1177, Jun. 2018.

[19] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, ‘‘Virtual adversarial training: A regularization method for supervised and semi–supervised learning,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[20] J. E. Tapiador, A. Orfila, A. Ribagorda, and B. Ramos, ‘‘Key–recovery attacks on KIDS, a keyed anomaly detection system,’’ IEEE Trans. Dependable Secure Comput., vol. 12, no. 3, pp. 312–325, May 2015.