DATA POISON DETECTION SCHEMES FOR DISTRIBUTED MACHINE LEARNING

Afzal Sayed
PG Scholar, Department of Computer Engineering,
Terna College of Engineering,
Osmanabad, Maharashtra, India - 413501
Guide - Professor Sujata Gaikwad
Hod - Professor Sujata Gaikwad
Email: afzalsayed711@gmail.com

Abstract: Challenges in DML systems are described in the project abstract. It talks about the problems and solutions in the context of DML settings. Businesses can gain from big data, but there is also a risk of DML. Distributed DML can be more attack oriented than Non-distributed. The project classifies DML into two types basic-DML and semi-DML. Basic-DML: Learning tasks are distributed across the components and done by them. Semi-DML not only assigns tasks to distributed resources, but also allocates additional resources to learn the dataset in the central server. However, in this research, we provide a novel approach to detecting data poisoning in basic-DML. It is a cross-learning mechanism to identify and reduce the poisoned data. Moreover, it presents a mathematical framework to find the optimum number of training iterations to achieve more accuracy. The study proposes an improvement to the data poison detection system for semi-DML. This optimises the resource allocation to learn the dataset efficiently utilising the resources of the system. You want to be more precise . . . In such cases you don't want any wastage.

Index terms *– Distributed machine learning, data poison detection, resource allocation.*

1. INTRODUCTION

DML has been widely used in the distributed system [1, 2]. There is no single node that can get the intelligent conclusion from a big dataset within an acceptable period [3]–[6]. A typical DML system [7] has a central server with a huge quantity of data available to it. It divides the data set into multiple parts and sends them to multiple workers for training, and the training results are returned to the center [8]–[10]. Lastly, the center can make use of these findings to create the ultimate model. Unfortunately, with the increase of distributed workers, it is difficult to ensure the security of each worker. This lack of security will increase the likelihood that the data will become compromised and will impact the training results. Poisoning assault [11]–[13] is a typical approach to manipulate the training data in machine learning.

In particular, when the newly created datasets are provided periodically to the dispersed workers for updating the decision model, the attacker will have more opportunities to poison the datasets, posing a more serious danger to DML. This limitation of the ML has garnered huge attention from the researchers. Dalvi et al. [14] demonstrated that when the attacker has complete information, he/she can tamper the data to trick the data miner. Later, Lowd and Meek [15] noted that it is not realistic to assume perfect information and developed attacks where the

attackers have some knowledge. Subsequently, a number of works were carried out [16]–[23], in a non-distributed ML setting.

A few relatively new projects are on the task of preventing data manipulation by DML. For example, Zhang and Zhu [24] developed a safe approach for DSVM based on game theory; and Esposito et al. [25] proposed a secure algorithm for collaborative deep learning based on game theory. These schemes are developed to be specific to a specific DML algorithm and have no application to the general DML case. The adversarial attack can fool many ML algorithms, necessitating the study of a widely applicable DML defence mechanism. To classify the DML, we considered the contribution of the centers in the dataset training tasks, and divided it into the basic-DML and semi-DML types. Then we present data poison detection algorithms for basic-DML, and semi-DML, respectively. The experimental results verify the effect of our proposed schemes.

2. LITERATURE SURVEY

MEC is one possible scenario that can meet the requirements of users for low latency applications. The thorough integration of multi-access technologies and MEC can greatly improve the access capacity between heterogeneous devices and MEC platforms[1]. However, the typical MEC network design cannot be directly utilised in the IoV due to the high speed mobility and intrinsic characteristics [10]. In addition, as there are numerous resource-rich vehicles on the road, it is a new chance to do task offloading and data processing on smart vehicles. In this paper, the authors first introduce a vehicular edge multi-access network based on considering automobiles as edge compute resources to construct a cooperative and distributed computing architecture, which will be beneficial for the cooperative and distributed computing architecture for the good merging of MEC technology into IoV. For immersive applications, co-located vehicles have the intrinsic features of collecting large amount of same and similar compute jobs. The collaborative task offloading and output transmission technique is developed to ensure low latency and application level performance. Last but not least, we discuss an example case study to offer suggestions for network framework design, 3D reconstruction. Numerical findings show that the suggested approach can lower the perceptual reaction time, while assuring the application level driving experiences.

The IoT platform is an integral part of improving the safety and efficiency of road transport systems by enabling ubiquitous connectivity via wireless communication among intelligent cars. But this IoT [2] concept requires an enormous spectrum resource requirement due to the requirement of constant connection and monitoring. CR is a promising approach to address the spectrum shortage problem by opportunistically using spectrum holes. But there are a number of challenges that need to be addressed for CR-based vehicle networks, including the high dynamics of the topology and time-varying states of the spectrum [12]. Furthermore, there are other challenges for efficient transmission scheduling, such as different vehicular communication modalities (vehicle to infrastructure, vehicle to vehicle, etc.) and data QoS requirements. Enlightened by this, we suggest a deep Q-learning based method to develop an optimum data transmission scheduling strategy in cognitive vehicular networks to reduce the data transmission cost completely utilizing all the

communication modalities and resources. Furthermore, we analyse the features of communication modes and spectrum resources selected by cars in different network states, and offer an efficient learning approach to derive optimal scheduling strategies. The performance of the proposed scheduling systems [2, 12] is presented in numerical results.

TensorFlow is a ML system to handle large-scale distributed and heterogeneous environments. TensorFlow encodes computation and shared state as dataflow graphs, and operations that change the state as operations on those graphs. It distributes the nodes of a dataflow graph across a cluster of many machines and across several processing devices on each machine such as multicore CPUs, general-purpose GPUs, and custom-designed ASICs called TPUs. This design gives the app developer the flexibility to try out different optimisations and training algorithms [17, 22, 23, 24] that involve shared state; in previous “parameter server” designs, the control of shared state is embedded in the system. The focus of TensorFlow is training and inference on deep neural networks, and it's appropriate for a variety of applications. TensorFlow is used in production in many of Google's services, is open-sourced and is also extensively used in ML research. This study [3] introduces the TensorFlow dataflow model, showcasing the remarkable results it can attain in actual world applications.

MXNet is an ML package supporting many languages to facilitate the creation of ML algorithms [17, 22, 23, 24], especially for deep neural networks. It has been deeply embedded in the host language, and is tightly coupled with declarative symbolic expressions and imperative tensor computation. It provides auto differentiation to obtain gradients. It is computation- and memory-efficient, and can be deployed on a wide range of hardware platforms including mobile devices, large-scale distributed GPU clusters. A study [4] talks about the design of the API of MXNet and the implementation of the system, and demonstrates how the symbolic expression and tensor operation are unified by embedding. We have obtained promising results in our preliminary experiments with large scale multiple GPU processors and deep neural networks.

The ML is always unleashing its strength in numerous applications. It has been brought to the fore in recent years partially due to the advent of big data. Better is never promised and challenged ML algorithms [7, 16] are involved in the context of big data. While the key benefit of big data is the capability of ML algorithms to find out much more subtle patterns and make much more timely and precise predictions than ever before, it also brings key challenges for ML including model scalability or distributed computing. This paper [5] outlines a framework of MLBiD to provide a basis for discussing the advantages and problems. It is focused on the orientation of the framework on the phases of pre-processing, learning and evaluation for the context of ML. In addition, there are another 4 parts of the framework: big data, user, domain and system. The phases of ML [16] and the components of MLBiD give guidance for identifying connected opportunities and difficulties and open for future work in various unknown or under researched research areas.

3. METHODOLOGY

i) Proposed Work:

The proposed system combines Data Poison Detection techniques with distributed learning techniques like Basic DML and Semi DML. The working principle of the system is to have various worker nodes and one central server work together to detect and eliminate poisoned data from the dataset. The system also aims to assess and compare the aforementioned distributed learning methods in improving the accuracy of the ML model. The method is effective in recognising and eliminating poisoned data, and it enhances the integrity of both basic-DML and semi-DML cases. Basic-DML also achieves improved results in identifying and correcting poisoned data with its cross-learning and mathematical modelling. The system offers a resource allocation method in semi-DML that decreases resource wastage and ensures effective resource utilisation. gains in basic-DML result in substantial gains in the accuracy of the final model, showing that the approach is effective in improving ML outcomes.

ii) System Architecture:

In this paper, we classified DML into basic-DML and semiDML accordingly as shown in Fig.1. Both of the two scenarios have a center which is composed of database, compute server and parameter server [3, 4, 5, 7]. However, there are different roles played by the center in the two cases. FIGURE 1. In basic-DML, the center will not have free computing power to train sub-datasets and will pass all sub-datasets to the distributed workers. In the basic-DML, as a result, the center merely needs to gather training outcomes coming from distributed workers by using the parameter server. In the semi-DML scenario, on the other hand, the center is equipped with some spare resources in the computing server to learn the sub-datasets. It will retain some sub-datasets and learn from it by itself. In the semi-DML, therefore, the center will be learning from some of the sub-datasets as well as incorporating the results from the center and distributed workers.

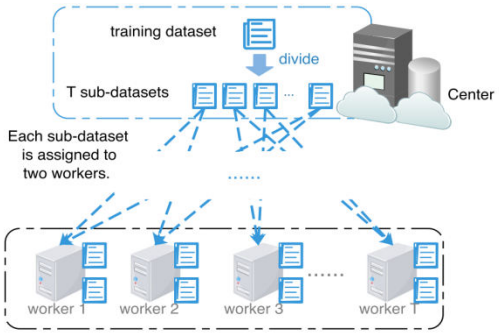

Fig 1 Proposed Architecture

Since basic-DML system with cross-learning mechanism may have varied number of training loops as mentioned in Section III.B. In this part, we will discuss the influence of the cross-learning mechanisms with various training loops. We reflect the influence in different scenarios by the PFT. The PFT of the proposed strategy is the probability that the attacker does not impact the final trained model when randomly attacking y workers from x workers. In the virtual topology, two workers are neighbours if they have the same sub-dataset. Once successfully compromised by the attacker, no two neighbouring workers can be distinguished. Thus, the PFT is equal to the likelihood that the attacker cannot pollute two neighbouring workers simultaneously. Now, we will discuss the PFT with various training loops and attempt to find the optimal number of training loops that results in the most effective suggested scheme.

iii) Modules:

1. In this module, user can upload a data set in the system. This is the first step where we upload the data-set in our case the 'heart.csv' file into the application.

2. This module splits the data into equal pieces to be distributed among different worker nodes, after the data is uploaded. The data set consists of 304 records, which is divided into two halves of the size 152 each in this project.

3. This module then sends the fragments of this dataset to worker nodes when they are separated. Then the worker nodes run the Basic-DML technique to create the models based on ML and calculate accuracies to compare with.

4. The first one in the Semi-DML procedure. This enables the central server to assign resources, run Semi-DML, perhaps to improve the ML model further, and to assess its accuracy.

5. Finally, this module generates a graph which compares the accuracy of several algorithms. It draws the curve of accuracies of the current SVM and the accuracies achieved after applying Basic-DML and Semi-DML methods, which allows a visual comparison of the results [25].

4. EXPERIMENTAL RESULTS

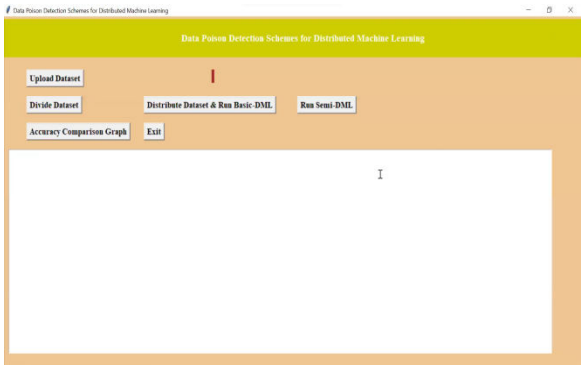

Fig 2 Home screen

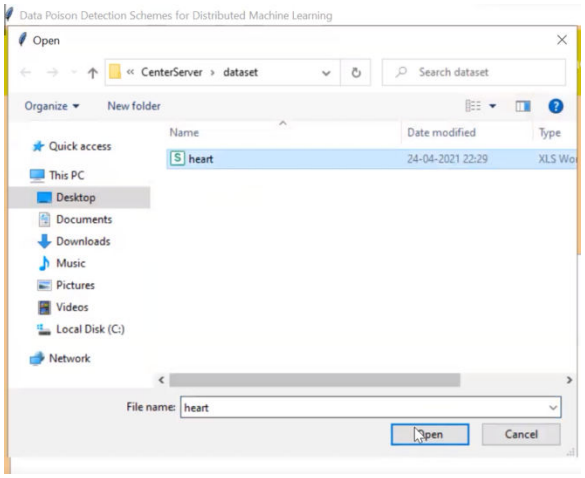

Fig 3 Upload dataset

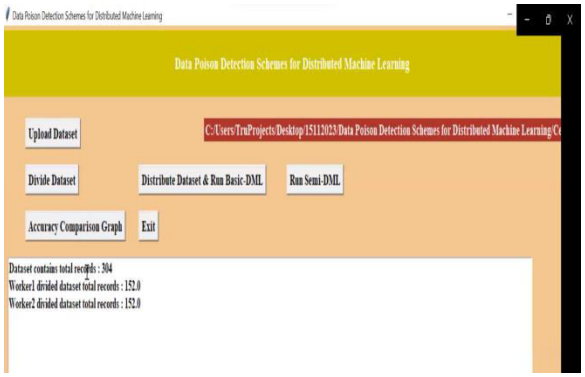

Fig 4 Divide dataset

Fig 5 Distribute dataset & Run basic DML

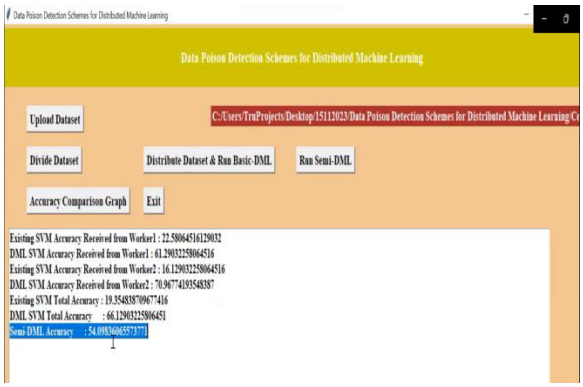

Fig 6 Run semi DML

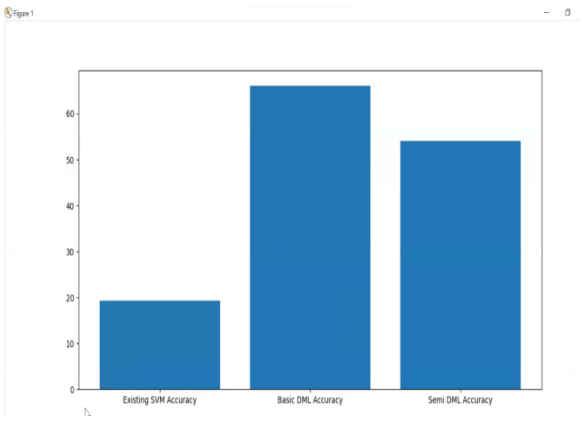

Fig 7 Accuracy comparison graph

5. CONCLUSION

The effectiveness of the system in identifying and removing inaccurate or inaccurate information resulted in significant improvement in ML capabilities in the dispersed environment. The higher accuracy indicates how effective the Data Poison Detection method has been in enhancing the models. Both Basic-DML and Semi-DML fared better than the conventional SVM [25]. In this clear presentation, the important achievement of integrating DPD techniques during the distributed learning process is highlighted. The technology demonstrated its scalability to data distributedly. The use of Basic-DML and Semi-DML methods on various datasets highlighted their adaptability and efficiency, as they successfully applied to various data sets.Their adaptability and efficiency were showcased when they applied Basic-DML and Semi-DML methods to various datasets. The central server actively participated in the Semi-DML, allocated resources to the training process, so that good accuracy results were obtained [2, 3, 4, 5]. This shows that effectively allocating resources is essential to get the best performance from the model.

6. FUTURE SCOPE

Future research should look at more complex detection algorithms rather than Data Poison Detection [15, 18]. These approaches would reinforce the system against new and evolving attack strategies, assuring continued accuracy and security. Adaptive learning models which change dynamically in response to irregularity detected in the system could be useful to the system. This continuous enhancement would render the system more resilient to data poisoning attacks and equip it to counteract emerging threats in the future. Real-time monitoring tools can be used to detect and counteract data poisoning in real-time. This proactive strategy helps assure continuous system security by addressing possible risks as soon as they arise. Advanced optimisation algorithms can be used to further reduce computing efficiency without reducing accuracy. These algorithms will enable faster processing of bigger data volumes, which will improve the overall performance of the system.

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 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-GonzÆlez, 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,'' 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.

[21] M. Kantarcioglu, B. Xi, and C. Clifton, ‘‘Classifier evaluation and attribute ˇ selection against active adversaries,’’ Data Mining Knowl. Discovery, vol. 22, nos. 1–2, pp. 291–335, Jan. 2011.

[22] S. Rota Bulo, B. Biggio, I. Pillai, M. Pelillo, and F. Roli, ‘‘Randomized prediction games for adversarial machine learning,’’ IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 11, pp. 2466–2478, Nov. 2017.

[23] N. Baracaldo, B. Chen, H. Ludwig, and J. A. Safavi, ‘‘Mitigating poisoning attacks on machine learning models: A data provenance based approach,’’ in Proc. 10th ACM Workshop Artif. Intell. Secur., 2017, pp. 103–110.

[24] R. Zhang and Q. Zhu, ‘‘A game–theoretic approach to design secure and resilient distributed support vector machines,’’ IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 11, pp. 5512–5527, Nov. 2018.

[25] C. Esposito, X. Su, S. A. Aljawarneh, and C. Choi, ‘‘Securing collaborative deep learning in industrial applications within adversarial scenarios,’’ IEEE Trans. Ind. Inf., vol. 14, no. 11, pp. 4972–4981, Nov. 2018.

[26] B. Efron, ‘‘Bootstrap methods: Another look at the jackknife,’’ Ann. Statist., vol. 7, pp. 569–593, Jan. 1979.

[27] C. Molnar. (2019). A Guide for Making Black Box Models Explainable. https://christophm.github.io/interpretable-ml-book/

[28] M. Li and I. Sethi, ‘‘Confidence-based active learning,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 8, pp. 1251–1261, Aug. 2006.

[29] B. Zhou, X. Tang, H. Zhang, and X. Wang, ‘‘Measuring crowd collectiveness,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 8, pp. 1586–1599, Aug. 2014.

[30] X. Gong, Q. Yao, M. Wang, and Y. Lin, ‘‘A deep learning approach for oriented electrical equipment detection in thermal images,’’ IEEE Access, vol. 6, pp. 41590–41597, 2018.

[31] L. Yann, C. Corinna, and J. B. Christopher. (2013). The Mnist Database of Handwritten Digits.

[Online]. Available: http://yann.lecun.com/exdb/mnist/