DATA POISON DETECTION SCHEMES FOR DISTRIBUTED MACHINE LEARNING

MR. R. RAMESH 1, MANNURU MAHESH2, KASIREDDI RAMESH 3,

1Department of Information Technology, Dr. M.G.R Educational and Research Institute, Chennai-600095.

India, ramesh.it@drmgrdu.ac.in

2Department of Information Technology, Dr. M.G.R Educational and Research Institute, Chennai-600095.

India, rameshkasireddi23@gmail.com

3Department of Information Technology, Dr. M.G.R Educational and Research Institute, Chennai-600095.

India, e38maheshchowdary@gmail.com

ABSTRACT

Distributed machine learning (DML) can achieve large dataset training when a single node cannot obtain accurate results in an acceptable time. However, compared to non-distributed environments, there are inevitably more potential targets for attackers. This article classifies DML into basic DML and semi-DML. In Basic DML, a central server distributes learning tasks to distributed machines and aggregates the learning results. In Semi-DML, the center server spends additional resources on learning the dataset in addition to his tasks in Basic DML. We first proposed a novel data poisoning detection scheme for basic DML that uses a cross-learning mechanism to find poisoned data. We prove that the proposed mutual learning mechanism generates training loops, based on which a mathematical model is built to find the optimal number of training loops. We then introduce an improved data poisoning detection scheme for Semi-DML that uses centralized resources to provide better learning protection. Optimal resource allocation approaches are developed to efficiently use system resources. Simulation results show that the proposed scheme can significantly improve the accuracy of the final model by up to 20% for support vector machines and up to 60% for logistic regression in basic DML scenarios. Moreover, in semi-DML scenarios, an improved data poisoning detection scheme with optimal resource allocation can reduce wasted resources by 20-100%.

Keywords: DML, CR, IOT, MXN, SVM, DSVM, Q learning.

1. INTRODUCTION

Distributed machine learning (DML) is widely used in distributed systems where a single node cannot make intelligent decisions from huge datasets in an acceptable time [1]. In a typical DML system, a huge amount of data is available on a central server. It splits the dataset into different parts and distributes them to distributed workers, who perform training tasks and send the results back to the center [2]. Finally, the center integrates these results and outputs the final model.Unfortunately, with the increasing number of distributed employees, it is difficult to ensure the safety of all employees. This lack of security increases the risk that attackers can tamper with datasets and manipulate training results. Poisoning attacks are a typical way to manipulate training data in machine learning. Particularly in scenarios where

newly generated records need to be sent periodically to a distributed workforce to update decision-making models, there is a high chance that an attacker will tamper with the records, thereby increasing the threat in DML. becomes larger.Such vulnerabilities in machine learning have received significant attention from researchers. Dalvi et al. [3] showed for the first time that an attacker can manipulate data to defeat data miners if they have complete information. Next, Lowd et al. [4] argued that the assumption of perfect information is unrealistic and proved that an attacker can use a piece of information to construct an attack. Subsequently, a series of studies focused on the context of non-distributed machine learning were conducted. Recently, several efforts have been made to prevent data manipulation in DML. For example, Zhang et al. and Esposito et al [5]. used game theory to design a distributed support vector machine (DSVM) or secure algorithm for collaborative deep learning. However, these schemes are designed for specific DML algorithms and cannot be used in general DML situations. Adversarial attacks can mislead various machine learning algorithms, so there is an urgent need to consider generally applicable DML protection mechanisms.In this article, we classify DML into basic distributed machine learning (Basic-DML) and semi-distributed machine learning (Semi-DML) depending on whether centers share resources for dataset training tasks. Next, we introduce basic DML and quasi-DML data poisoning detection schemes, respectively [6]. Experimental results confirm the effectiveness of the proposed scheme. The main contributions of this paper are summarized below.We proposed a basic DML data poisoning detection scheme based on the so-called mutual learning data mapping mechanism. Therefore, we prove that the mutual learning mechanism generates training loops and provides a mathematical model for finding the optimal number of training loops with the highest security.We present a practical method for identifying anomalous training results that can detect contaminated datasets at a reasonable cost.For Semi-DML, we propose an improved data poisoning detection scheme. Which one can provide better learning protection. An optimal resource allocation scheme is developed to utilize system resources efficiently.

2. LITERATURE SURVEY

TITLE: Collaborative task offloading in vehicular edge multi-access networks

G. Qiao, S. Leng, K. Zhang, and Y. He,

Mobile edge computing (MEC) has emerged as a promising paradigm for realizing user needs in low-latency applications. The tight integration of multi-access technology and MEC significantly increases the access capacity between disparate devices and MEC platforms. However, due to high-speed mobility and inherent characteristics, traditional he MEC network architecture cannot be directly applied to Internet of Vehicles (IoV). With more resource-intensive vehicles on the road, new opportunities are also emerging to outsource tasks and data processing to intelligent vehicles. To enable a proper convergence of MEC technologies in IoV, this paper first introduces a vehicular edge multi-access network that treats vehicles as edge computing resources and builds a collaborative and distributed computing architecture. . For immersive applications, co-located vehicles have the unique property of collecting identical and similar computational tasks that are important. We propose a cooperative mechanism for task offloading and output forwarding to guarantee low latency and application-level performance. Finally, we consider 3D reconfiguration as an example scenario that provides insight into the design of network frameworks. Numerical results show that the proposed scheme can reduce the

perceived response time while ensuring the driving experience at the application level.

TITLE: **Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks**

K. Zhang, S. Leng, X. Peng, L. Pan, S. Maharjan, and Y. Zhang,

Internet of Things (IoT) platforms have played a key role in improving the safety and efficiency of road transportation through the ubiquitous connectivity of intelligent vehicles through wireless communications. However, such an IoT paradigm places a heavy burden on limited spectrum resources due to the need for continuous communication and monitoring. Cognitive radio (CR) is a potential approach to alleviate the spectrum scarcity problem by opportunistically utilizing underutilized spectrum. However, the highly dynamic topology and time-varying spectrum conditions in CR-based vehicular networks pose several challenges that need to be overcome. Additionally, there are various vehicle communication modes such as: B. Vehicle-to-infrastructure, vehicle-to-vehicle, and data QoS requirements pose significant challenges in efficient transmission planning. Based on this motivation, this paper applies a deep Q-learning approach to design an optimal scheduling scheme for data transmission in cognitive vehicular networks while making full use of different communication modes and resources, reducing the transmission cost. minimize. Furthermore, we study the characteristics of communication modes and spectrum resources selected by vehicles in different network states and propose an efficient learning algorithm to obtain the optimal scheduling strategy. Numerical results are presented to demonstrate the performance of the proposed scheduling scheme.

TITLE: **Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems**

T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang

MXNet is a multilingual machine learning (ML) library that facilitates the development of ML algorithms specifically for deep neural networks. It is embedded in the host language and combines declarative symbolic representation with imperative tensor computation. Automatic differentiation is used to derive color gradation. MXNet is compute- and memory-efficient and runs on a variety of heterogeneous systems, ranging from mobile devices to distributed GPU clusters.This document describes both the API design and system implementation of MXNet, and how it handles both symbolic expression and tensor operation embeddings in a unified way. Our preliminary experiments have shown promising results in large-scale deep neural network applications using multiple GPU machines.

3. EXISTING SYSTEM

Unfortunately, with the increasing number of distributed employees, it is difficult to ensure the safety of all employees. This lack of security increases the risk that attackers can tamper with datasets and manipulate training results. Poisoning attacks are a typical way to manipulate training data in machine learning. Particularly in scenarios where newly generated records need to be sent periodically to a distributed workforce to update decision-making models, there is a high chance that an attacker will tamper with the records, thereby increasing the threat in DML. becomes larger.Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression tasks. However, it is mainly used in classification problems. The SVM algorithm represents each data item as a point in an n-dimensional space, where n is the number of features present. The value of each feature is the value at a particular coordinate. Next,

perform the classification by finding a hyperplane that clearly distinguishes the two classes.

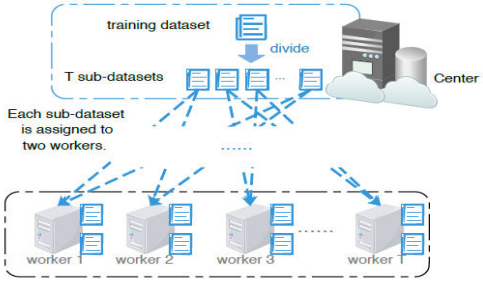
Disadvantages

In a distributed environment, an attacker may modify the training data and subsequently build the ML model. This model gives wrong results

4. PROPOSED SYSTEM

Convert DML to Basic Distributed Machine Learning (Basic-DML) and Semi-Distributed Machine Learning (Semi-DML) depending on whether the centers share resources for dataset training tasks. Next, we introduce basic DML and quasi-DML data poisoning detection schemes, respectively. Experimental results confirm the effectiveness of the proposed scheme.We classify DML into basic DML and semi-DML as shown in Figure 1. Both scenarios have centers containing databases, compute servers, and parameter servers. However, the center provides different functionality in these two scenarios. In a basic DML scenario, the center has no free computing resources for training subdatasets, and all subdatasets are sent to distributed staff. Therefore, the center simply integrates the training results of the distributed employees into the basic DML via the parameter server.

SYSTEM ARCHITECTURE



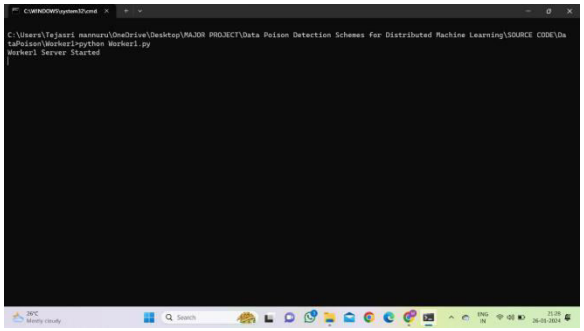
Mutual Learning Mechanism and Training Loop

The mutual learning mechanism provides the basis for creating backups of partial data sets and identifying contaminated partial data sets, as shown in the figure. In Section II, we showed that there are T workers and the data center partitions the training dataset into T (T ∈ N) subsets. In this mechanism, each partial dataset is assigned to her two workers, who produce the corresponding training results. For example, subdataset Di is assigned to workers ea and eb (a, b ∈ {1, ...). ,T}). The two workers produce two training results wia and wib, both corresponding to Di. Therefore, there are two training results corresponding to each subdataset. The algorithm of mutual learning mechanism is shown in Algorithm 2. If two workers receive the same partial data set, they are considered to have a virtual connection. Subdatasets are randomly assigned to different workers, so workers can have different virtual connections depending on their assignment. To abstract these connections between workers, this part introduces a virtual topology. In a virtual topology, a connection exists if there is a virtual connection between two workers (receiving the same partial data set). L = {l|i, j ∈ {1, . . , T }} denotes the set of all links in the virtual topology.

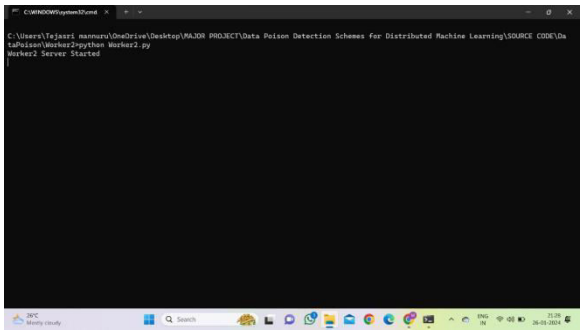
5. RESULT

It takes the existing SVM accuracy results from two worker nodes and suggests DML accuracy. In the screen above, you can see that the accuracy of the existing SVM is 19% when there is data poisoning in the dataset. After removing data poisoning using the DML technique, the curacy is at 51°C and clicks the "Run Semi-DML" button to allow the center server to provide resources for his DML and removes it from the dataset. Calculate by removing poisoning. accuracy.

To run the project, first start the Worker 1 node by double-clicking the run.bat file in the Worker1 folder and moving it to the bottom of the screen.



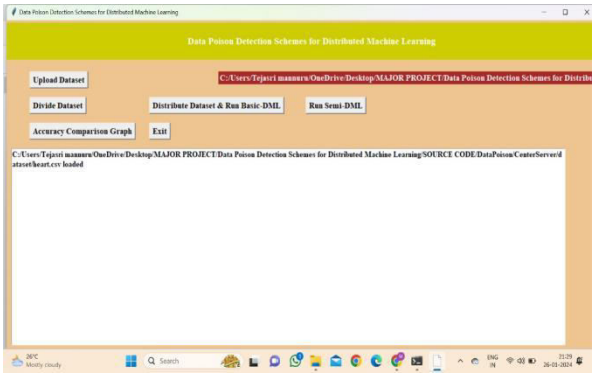
In the screen above, the Worker 1 server is started, and Worker 2 is started by double-clicking the run.bat file in the Worker2 folder.



In the screen above, the Worker2 server is running. Double-click the run.bat file in the Center Server folder to start the distributed server and proceed to the next screen.

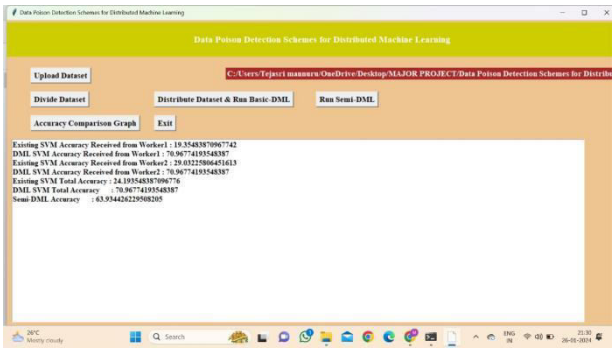
On the above screen, click the "Upload Dataset" button to upload and retrieve the dataset. Bottom of screen

On the screen above, select and upload the heart.csv file. Next, click the "Open" button to load the dataset and move to the screen below.



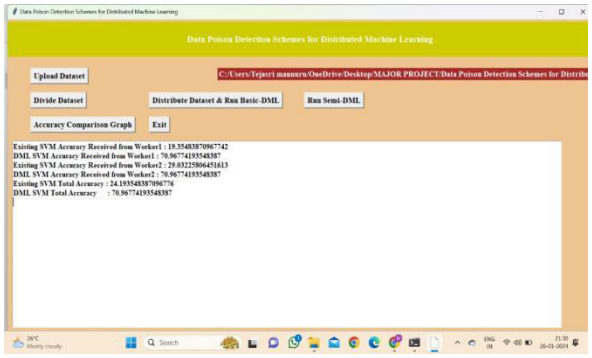
In the screen above, the dataset has been loaded. Click the Split Record button to split the record into 2 eqaul parts.

In the screen above, the data set contains 304 records, split evenly into two parts. Next, click the Distribute Dataset and Run Basic DML button to distribute the dataset to her two workers and get the accuracy results.

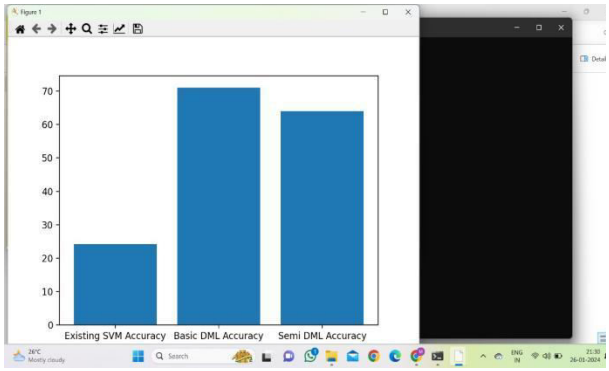


In the screen above, we are retrieving results from two worker nodes for the existing SVM accuracy and suggesting DML accuracy. In the screen above, you can see that the accuracy of the existing SVM is 19% when data poisoning is present in the dataset. After removing data poisoning using the DML technique, we obtained an accuracy of 51%. Now click on the Run

Semi-DML button to allow it. The center server provides resources for DML, removes poison from the dataset, and calculates 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

This article described data poisoning detection schemes in both basic and quasi-DML scenarios. The data poisoning detection scheme for basic DML scenarios uses parameter thresholds to find corrupted sub datasets. Additionally, we built a mathematical model to analyze the probability of finding a threat with different numbers of training loops. Furthermore, we presented an improved data poisoning detection scheme and optimal resource allocation in quasi-DML scenarios. Simulation results show that the proposed scheme can improve model accuracy for support vector machines by up to 20% in basic DML scenarios and by up to 60% in logistic regression. For semi-DML scenarios, the improved data poisoning detection scheme with optimal resource allocation can reduce resource wastage by 20-100% compared to the other two schemes without optimal resource allocation .

REFERENCES

1. 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..

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 of Things Journal, vol. 6, no. 2, pp. 1987–1997, 2019.

3. 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, "Tensor flow: A system for large-scale machine learning." in 12thUSENIXSymposium on Operating Systems Design and Implementation (OSDI), vol. 16. USENIX Association, 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: Aflexible and efficient machine learning library for heterogeneous distributed systems," CoRR, vol. abs/1512.01274, 2015..

5. Prasadu Peddi (2021), "Deeper Image Segmentation using Lloyd's Algorithm", ZKGINTERNATIONAL, vol 5, issue 2, pp: 1-7..

6. S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for bigdata: Asurvey," IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 531–549, 2017.

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." in11thUSENIX Symposium on Operating Systems Design and Implementation (OSDI), vol. 14. USENIX Association, 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 Network, vol. 30, no. 1, pp. 6–10, 2016.

9. Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home m2m networks: Architectures, standards, and qos improvement," IEEE Communications Magazine, vol. 49, no. 4, pp. 44–52, 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 Network Magazine, vol. 33, no. 3, pp. 10–17, 2019.11. Charmaz K (2016). Constructing grounded theory: A practical guide through qualitative analysis. Thousand Oaks: Sage

11.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.

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.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.

14.D. Lowd and C. Meek, ''Adversarial learning,'' in Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2005, pp. 641–647.

15.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.