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: Distributed Machine Learning (DML) enables efficient training over large-scale datasets but also introduces significant security challenges. Due to its decentralized nature, DML systems are more vulnerable to malicious attacks, particularly data poisoning. This work addresses these challenges by proposing effective detection mechanisms tailored for distributed environments.

We categorize DML into two types: Basic Distributed Machine Learning (Basic-DML) and Semi Distributed Machine Learning (Semi-DML). In Basic-DML, worker nodes independently process assigned data, whereas in Semi-DML, the central server also participates in the training process using additional computational resources.

To mitigate data poisoning in Basic-DML, a cross-learning based detection mechanism is introduced, which enhances the identification of malicious data. Additionally, a mathematical framework is developed to determine the optimal number of training iterations for improved detection accuracy. For Semi-DML, an enhanced detection approach is proposed that focuses on efficient resource allocation and improved learning performance. Experimental observations demonstrate that the proposed techniques significantly improve model accuracy while reducing the impact of poisoned data.

Keywords— Distributed Machine Learning, Data Poisoning, Security, Resource Optimization

INTRODUCTION

Distributed Machine Learning (DML) has become an essential approach for handling large-scale data processing tasks across multiple computing nodes. In traditional centralized systems, processing massive datasets within a limited time is often impractical. DML overcomes this limitation by dividing the dataset into smaller portions and distributing them across multiple worker nodes for parallel processing.

In a typical DML architecture, a central server is responsible for managing the dataset and distributing it among worker nodes. Each worker performs local training on its assigned subset of data and returns the results to the central server, which then aggregates them to produce the final model.

However, as the number of distributed workers increases, ensuring system security becomes more challenging. The presence of multiple nodes creates opportunities for attackers to inject malicious or manipulated data into the training process, commonly known as data poisoning attacks. These attacks can significantly degrade model performance and lead to incorrect predictions.

Earlier studies have demonstrated that adversaries can manipulate training data even with limited system knowledge. While several defense mechanisms have been proposed, most are designed for centralized

machine learning systems and are not directly applicable to distributed environments.

Recently, some research efforts have focused on securing DML systems using techniques such as game theory. However, these approaches are often limited to specific algorithms and lack general applicability.

To address these limitations, this work introduces a generalized framework for detecting data poisoning in distributed environments. We classify DML into Basic-DML and Semi-DML based on the role of the central server in the training process. For both categories, dedicated data poisoning detection mechanisms are proposed. The effectiveness of these approaches is validated through experimental analysis.

Recently, there area couple of efforts devoted in preventing data from being manipulated in DML. For example, Zhang and Zhu [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. In this paper, we classify 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.

1. LITERATURE SURVEY

Recent advancements in distributed computing and intelligent systems have significantly influenced the evolution of distributed machine learning (DML). One such paradigm is Mobile Edge Computing (MEC), which enables low-latency processing by bringing computational resources closer to end devices. The integration of MEC with heterogeneous access technologies has improved communication efficiency; however, its direct application in vehicular environments remains challenging due to high mobility and dynamic network conditions.

To address these issues, vehicular edge computing frameworks have been proposed, where vehicles themselves act as computational nodes. These frameworks enable collaborative task execution and efficient data sharing among nearby vehicles. For instance, cooperative task offloading mechanisms have been introduced to reduce latency and improve application-level performance, particularly in computation-intensive scenarios such as 3D reconstruction.

Similarly, the Internet of Things (IoT) has played a crucial role in enabling intelligent transportation systems through continuous connectivity among devices. However, the increasing demand for communication in IoT-based vehicular networks introduces spectrum scarcity challenges. Cognitive Radio (CR) has emerged as a promising solution by allowing dynamic spectrum utilization. In this context, reinforcement learning techniques, such as deep Q-learning, have been employed to optimize data transmission scheduling, ensuring efficient utilization of available communication resources while minimizing transmission costs.

From a system perspective, large-scale machine learning frameworks such as TensorFlow and MXNet have enabled scalable and efficient training across distributed environments. These frameworks support heterogeneous computing architectures, including CPUs, GPUs, and specialized hardware accelerators. Their flexible design allows developers to experiment with various optimization strategies and training algorithms, making them suitable for large-scale deep learning applications.

Furthermore, the emergence of big data has significantly expanded the scope of machine learning. While large datasets enable more accurate and robust models, they also introduce challenges related to scalability, distributed processing, and system complexity. To address these challenges, structured frameworks have been proposed that integrate data

preprocessing, learning, and evaluation phases with system-level considerations.

Despite these advancements, security remains a critical concern in distributed machine learning systems. Existing research primarily focuses on centralized environments or algorithm-specific defenses, leaving a gap in generalized protection mechanisms for DML. This highlights the need for robust and scalable data poisoning detection techniques tailored specifically for distributed architectures.

2. METHODOLOGY

i) Proposed Work:

The proposed framework incorporates data poisoning detection mechanisms into distributed machine learning architectures, specifically targeting both Basic-DML and Semi-DML environments. The system consists of a central server and multiple distributed worker nodes that collaboratively participate in the training process.

The primary objective of the framework is to identify and eliminate malicious or corrupted data samples that may compromise model performance. In the Basic-DML setup, a cross-learning mechanism is employed, where multiple worker nodes validate each other's outputs to detect inconsistencies caused by poisoned data. This collaborative validation improves the reliability of the training process.

Additionally, a mathematical model is introduced to determine the optimal number of training iterations required to maximize detection accuracy while minimizing computational overhead. This ensures that the system maintains a balance between performance and efficiency.

In the Semi-DML setup, the central server actively participates in the training process by utilizing its computational resources. A resource allocation strategy is implemented to optimize workload distribution between the central server and worker nodes. This approach not only enhances model accuracy but also prevents unnecessary resource consumption.

ii) System Architecture:

The architecture of the proposed system is divided into two primary configurations: Basic-DML and Semi-DML.

In the Basic-DML configuration, the central server is responsible for dataset management and result aggregation. The dataset is partitioned into multiple subsets and distributed among worker nodes. Each worker independently trains a model using its assigned data and sends the results back to the central server. The server then aggregates these results to generate the final model.

In contrast, the Semi-DML configuration extends this architecture by allowing the central server to participate in the training process. In addition to aggregating results, the server also trains on a portion of the dataset. This hybrid approach improves model robustness and enhances overall system performance.

A key component of the architecture is the cross-learning mechanism, which establishes relationships between worker nodes. Workers sharing similar data subsets are considered adjacent in a virtual topology. The effectiveness of the detection mechanism is evaluated using the Probability of Finding Threats (PFT), which measures the likelihood of successfully identifying poisoned data.

The analysis of PFT under varying training iterations helps determine the optimal configuration for maximizing detection efficiency while minimizing false negatives.

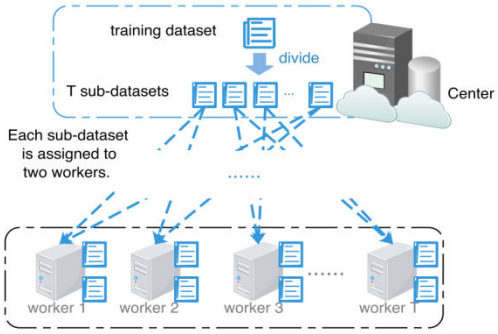

Fig 1 Proposed Architecture

Since a basic-DML system which using the cross-learning mechanism may have a different number of training loops, which is discussed in section III.B. Therefore, in this part, we will discuss the effect of cross-learning mechanisms with different training loops. We use the probability of finding threats (PFT) to reflect the effect in different situations. When the attacker randomly attacks y workers within x workers, the PFT of the proposed scheme is the probability when the attacker fails to influence the final trained model. In the virtual topology, two workers would be adjacent if they have the same sub-dataset. When the attacker successfully compromises two adjacent workers, our scheme cannot distinguish them. Therefore the PFT is equal to the probability that the attacker cannot simultaneously pollute two adjacent workers. In the following, we will discuss the PFT with different training loops and try to find the optimal number of training loops which make the proposed scheme most effective.

iii) Modules:

The system is composed of the following functional modules:

1. **Dataset Upload Module**
   This module allows users to upload datasets into the system. It serves as the entry point for initiating the distributed learning process.
2. **Dataset Partitioning Module**
   The uploaded dataset is divided into equal subsets to facilitate distributed processing. This ensures balanced workload distribution among worker nodes.
3. **Distributed Training Module (Basic-DML)**
   In this module, worker nodes perform independent training on their respective data subsets. The results are then transmitted to the central server for aggregation.
4. **Hybrid Training Module (Semi-DML)**
   This module enables the central server to participate in the training process alongside worker nodes. It incorporates resource allocation strategies to optimize computational efficiency.
5. **Performance Evaluation Module**
   This module compares the performance of different learning approaches. It generates visual representations, such as graphs, to illustrate accuracy improvements achieved through the proposed methods.

3. EXPERIMENTAL RESULTS

Fig 2 Home screen

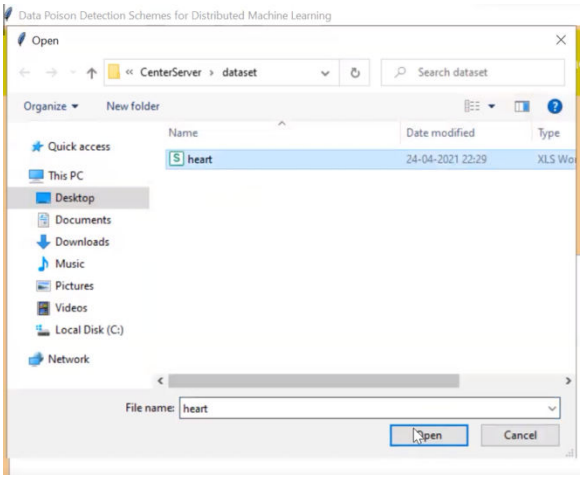

Fig 3 Upload dataset

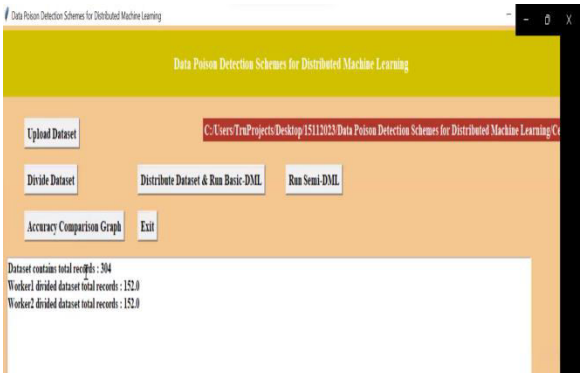

Fig 4 Divide dataset

Fig 5 Distribute dataset & Run basic DML

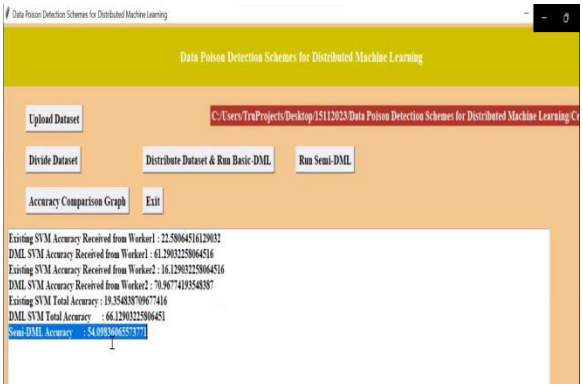

Fig 6 Run semi DML

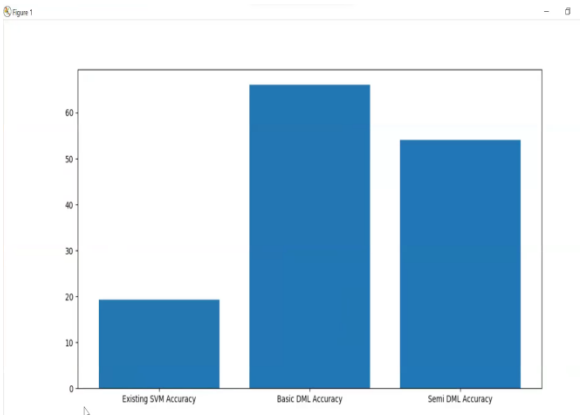

Fig 7 Accuracy comparison graph

4. CONCLUSION

This study presents a comprehensive framework for detecting and mitigating data poisoning attacks in distributed machine learning environments. By classifying distributed learning into Basic-DML and Semi-DML architectures, the work provides a structured approach to addressing security challenges inherent in decentralized systems.

The proposed cross-learning mechanism in Basic-DML effectively enhances the detection of malicious data by enabling collaborative validation among worker nodes. Additionally, the introduction of a mathematical model for optimizing training iterations ensures improved detection accuracy without incurring excessive computational overhead.

In the Semi-DML configuration, the active participation of the central server in the training process significantly improves model robustness and

accuracy. The implementation of an efficient resource allocation strategy further ensures optimal utilization of computational resources.

Experimental results demonstrate that the proposed approaches outperform traditional machine learning techniques, such as SVM, in terms of accuracy and resilience to data poisoning attacks. The findings highlight the importance of integrating security mechanisms directly into distributed learning frameworks.

Overall, this work contributes to the advancement of secure distributed machine learning by providing scalable, efficient, and generalized solutions for data poisoning detection.

5. FUTURE SCOPE

While the proposed framework demonstrates significant improvements in detecting data poisoning attacks, several opportunities exist for further enhancement and exploration.

Future research can focus on the development of **adaptive and self-learning detection mechanisms** that dynamically adjust to evolving attack patterns. Such systems can leverage advanced techniques such as reinforcement learning and meta-learning to continuously improve their detection capabilities.

Another promising direction is the integration of **blockchain technology** to ensure data integrity and secure communication among distributed nodes. Blockchain-based solutions can provide decentralized trust mechanisms, reducing the risk of malicious data injection.

Additionally, the incorporation of **real-time monitoring and anomaly detection systems** can enable immediate identification and mitigation of poisoning attacks. This would enhance the system's responsiveness and overall security.

Optimization of computational efficiency remains another key area for improvement. Advanced optimization algorithms and hardware acceleration techniques can be explored to reduce processing time while maintaining high detection accuracy.

Finally, extending the framework to support **heterogeneous and large-scale distributed environments**, such as federated learning systems, can further enhance its applicability in real-world scenarios.

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/