Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 99-105
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

# An Anti-Poisoning Attack Method for Distributed AI System

**Xuezhu Xin[1], Yang Bai[1], Haixin Wang[1], Yunzhen Mou[2], Jian Tan[2]**

[1]Innovation Center for Intelligent System on Recognition and Decision, Beijing Jinghang Research Institute of Computing and Communication, Beijing, China
[2]School of Digital Media and Art Design, Beijing University of Posts and Telecommunications, Beijing, China
Email: myz2017@bupt.edu.cn

**How to cite this paper:** Xin, X.Z., Bai, Y., Wang, H.X., Mou, Y.Z. and Tan, J. (2021) An Anti-Poisoning Attack Method for Distributed AI System. *Journal of Computer and Communications*, **9**, 99-105.
https://doi.org/10.4236/jcc.2021.912007

**Received:** October 12, 2021
**Accepted:** December 28, 2021
**Published:** December 31, 2021

## Abstract

In distributed AI system, the models trained on data from potentially unreliable sources can be attacked by manipulating the training data distribution by inserting carefully crafted samples into the training set, which is known as Data Poisoning. Poisoning will to change the model behavior and reduce model performance. This paper proposes an algorithm that gives an improvement of both efficiency and security for data poisoning in a distributed AI system. The past methods of active defense often have a large number of invalid checks, which slows down the operation efficiency of the whole system. While passive defense also has problems of missing data and slow detection of error source. The proposed algorithm establishes the suspect hypothesis level to test and extend the verification of data packets and estimates the risk of terminal data. It can enhance the health degree of a distributed AI system by preventing the occurrence of poisoning attack and ensuring the efficiency and safety of the system operation.

## Keywords

Data Poisoning, Distributed AI System, Credit Probability Mechanism, Inspection Module, Suspect Hypothesis Level

## 1. Introduction

The security of distributed AI systems is always a concern. As a large distributed network system, it has much information that should be frequently synchronized to take the real-time operation of the network such as frequent data transmission, data with terminal characteristics and adjusted or partially adjusted model [1]. However, it is also feasible for attackers to destroy and falsify data. From the view of Poisoning attacks in distributed AI systems, every client

has access to model parameters and training data, it is probable that some malicious clients will send tampered data or weights to the server, which can affect the global mode [2]. Poisoning attacks can generally be divided into three categories: Data Poisoning, Model Poisoning and Data Modification [3].

Traditional defense approaches for Poisoning attack fall into two categories: Proactive defense and Reactive defense [4]. Proactive defense is a way of preparing effective defense techniques on the basis of evaluating which threats to meet, while reactive defense is an instant defense operation performed only as some attack detected. However, these two types of defense approaches are more ideal classification than practical methods. It is difficult to determine which the threats from attackers are [5]. For example, a data packet transmitted normally can hardly be predicted as data poisoning or model poisoning according to its characteristics. If the criteria for predicting abnormal threat data are lowered, a large amount of data will be identified or discarded, which will result in low efficiency among distributed systems. If the criteria for predicting abnormal threat data are increased, a large amount of abnormal data will be mixed up with ordinary data packet in normal business processes, which will result in abnormal model and failure of business objectives.

In view of this problem, this paper establishes the suspect hypothesis level to test and extend the verification of data packets. It also estimates the risk of terminal data in distributed AI system by combining the historical credit record, data characteristics, data timestamp and some other data from the terminal and requires more evidence for screening. On the one hand, this algorithm can improve the overall defense efficiency of the system. On the other hand, it can guarantee the comprehensiveness of abnormal data identification.

## 2. Proposed Algorithm

The core technical problem solved by the proposed algorithm is how to make the data poisoning of AI distributed system be identified quickly and effectively. Conventional methods are usually hard to keep the balance of time consuming and algorithm performance [6]. Focusing too much on models and data anomalies will consume a lot of time and computing resources in selecting and discarding packets, which may contain a lot of normal data. Meantime, if the compatibility between data and model is too wide, camouflaged abnormal data and model cannot be recognized, which will lead to confusion in AI training, repeated divergence of AI model and failure to converge or even finally collapse of the training process.

This algorithm establishes a set of models and data identification methods with continuity and flexibility to solve this problem so that to improve the efficiency and accuracy of defense.

### 2.1. Common Senses

The algorithm is proposed based on some common senses.

1) Forgery data and model need a particular module [7] which will lead a result that they need to resident one fixed terminal since the overhead of frequent jumping installation and deployment is too high. If the hibernation gap attack is carried out, the attack frequency and data volume will be too small to damage.

2) Forged data and model can only be applied to data packets at one or several levels rather than running through the whole data link [8]. For example, if the photos taken by users are doped with forged data, it must be reflected in the final JPG file rather than in the RAW format of the sensor that does not need to be transmitted. On the one hand, the processing amount of data is large and the forging time is too long. On the other hand, the RAW format is usually not transmitted in network, which is meaningless for poison. Similarly, if terminal user interaction data is used to submitted, only user interaction data in certain interfaces or periods of time will be collected. Data poisoning can only modify the last sent interactive information packet. It is impossible to modify the original whole process data.

3) Forged data and model will lead to abnormal operation of the synchronized AI model [9]. Although specific data packets cannot be located, the reliability of the existence of poisoned data in an interval data packet is quite high.

## 2.2. Important Modules

Based on the above three common senses, the algorithm proposal establishes a distributed AI system active defense method that can accurately and efficiently identify the poison data and local model. The algorithm includes the following steps:

1) Credit probability mechanism [10]. Set the upper limit of toxic model or data submission quantity, and establish the ratio between the number of measured toxic data and the upper limit of max poison times as the overall credit probability rate of distributed nodes.

$$C_{node} = \frac{\sum_{time=start}^{time=now} \text{Poison Times}}{\text{Max Poison Times}} \tag{1}$$

2) Set accident responsibility interval and model mirroring time. The error of AI models will usually constantly reduce during training. When toxic data or toxic local models are mixed, the training of AI models will show significant regression or confusion. So, in case of the damage of the whole system, it is necessary to mirror the AI global model regularly. The time cycle of backup is related to the complexity and scale of AI model. Generally speaking, the mirror can be established according to the punch training of one or more data, the error function value decreases 10% or a fixed proportion compared with the error value of the last mirror.

The definition of poisoning also depends on the direction prediction of distributed AI system model update. To put it simply, the data and local model are assumed stable when the error enters the sequence of decreasing 10% or other

fixed proportion. At this time, if the error change stagnates or increases, it is considered as the appearance of poisoning. Then the data and local model increments (including data punch, local model punch and source) from the previous mirroring phase up to now will enter the risk pool, and each provider bears the risk of accidents.

The definition of risk quantification comes from the poisoning level, which is proportional to the $P_{degree}$ for the difference between the predicted model performance and the actual model performance. K is the normalization coefficient. The worse the performance is, the worse the poisoning is, and the greater the responsibility of poisoning assigned by each distributed node in the responsibility range will be. Specifically, since the responsibility of poisoning cannot locate which distributed node poisons or poisons the most, the arithmetic mean value of poisoning grade can be applied directly.

$$\begin{aligned} P_{degree} &= \kappa \times \left| E_c - E_p \right| \\ P_{node} &= \frac{P_1}{\sum nodes} \end{aligned} \tag{2}$$

3) Local cache module. The function of the local cache model is to reserve all the original data after the completion time of broadcast mirroring on the central node. It should be properly kept until the storage is released when the next central node image is broadcast and the next round of raw data caching is started. For distributed nodes that collect user interaction data, the local cache should store all original user input data and local intermediate data for each processing step.

4) Inspection module. The function of the inspection model is to poll the original data of each node according to the risk value. First, the inspection module is independent of any distributed AI system. It has the most comprehensive inspection rights on child nodes. Secondly, the inspection module does not need to keep activated. It only checks from high nodes to low nodes according to the risk probability of each sub-node after determining the poisoning. Third, inspection module's node check is evaluating the raw data and uploads midterm data processing. Specifically, check items includes the underived phenomenon on datalink, obvious fracture and data tampered. Inspection module retrieve all the local cache and network structure of the client node and simulate in its own processing container. Last, once the inspection module detects one of three characteristics of data poisoning, the terminal node of the customer will be removed from the system. According to the risk value, the offline time $Time_{offline}$ can be enlarged. $Time_{offline}$ is an adjustable multiple of $Time_{baseline}$, which is composed of the base $\gamma$ and geometric series. The geometric series includes the product of the number of poisonings and the severity of poisonings on this node in history. Obviously, the base must be greater than 1. The higher the number of poisonings or the confirmed severity of poisonings, the delisting time of this node will be more greatly delayed. This should also be regulated by the system administrator to make sure that vulnerable nodes that have been attacked repeatedly

could soon become banned indefinitely (for example being taken down for more than 10 years). It should also inform the client user to update the system or remove viruses. It is only after reaching the down time that the node could be allowed to reconnect to the network.

$$\text{Time}_{\text{offline}} = \text{Time}_{\text{baseline}} \times \gamma^{\text{Times}_{\text{history}} \times \sum_{n=1}^{n=i} \text{P}_{\text{degrees}}} \tag{3}$$

## 3. Experiment

The experiment scheme takes an AI distributed system based on Hadoop and Spark as an example. The core AI model of this distributed system training is a user category recognition model and the source data is the interactive data packets of each mobile terminal. Its training result is to automatically process user group aggregation with relatively uniform spatial distance. The credit probability base, which is also called the maximum number of poisons, is 100.

First of all, the difference between the new user interaction data and the center point of each user group is obvious after a period of stable operation. The new data is always close to the center point of one user type. Suppose the distance of them is d. Then the distance between it and the center point of other user groups will be more than 2 times of D.

Secondly, if there is forged data poisoning, new user interaction data will appear near the peripheral circle center of each user group center or far away from the center of all user groups. Because only at that condition the poisoning will cause invalid rollback of model training or invalid splitting of cluster center. In these two cases, the criterion for determining the poisoning is that the variance of the distance to the center of all user groups does not exceed 5% of the distance itself, which means the distance cannot be effectively classified or may even cause the failure of the group classification model. 5% - 4% is taken as grade one poisoning. 4% - 3% is taken as grade tow poisoning. 3% - 2% is taken as grade three poisoning. 2% - 1% is taken as grade four poisoning. 1% - 0% is taken as grade five poisoning.

Third, according to the algorithm proposed in this paper, the mirror of the central model submits user interaction data of 1 punch for each node. Since the processing is independent, the sharing of responsibility for determining poisoning becomes independent.

Fourth, the starting threshold of the inspection model is 1. When poisoning occurs for the first time in the initial operation, the credit probability is 0. Since only one node submits data between mirrors, when a node submits data and causes level 1 poisoning, the overall poisoning risk of the node will become 1 and the inspection can be carried out. Meantime, the inspection module retrieves the original interactive data between the two mirrors of the node and performs internal reasoning with the submitted data. Original interactive data includes user location and sliding time. If inspection module does not find original interactive data with significant regularity contact frequency but the sub-

mitted data is detected a significant rule, the middle of the data sampling filter will be obviously not even time or uniform area and can be determined as fake data. The node toxic poisoning can be identified.

Fifth, the inspection module calculates the offline time of the node as 192 hours, which means the user node is allowed to submit the certificate of virus detection and go online again after 8 days. The time is calculated based on the current poisoning level 3, the base $\gamma$ 2, the historical number of poisoning 1 and the basic offline time 24 hours. The overall credit probability of the node is 1%. If the poisoning event of the same level occurs again by this node, the number of historical poisoning records will be 2, which means the user node is allowed to go online after 64 days and the overall credit probability of the node is 2%. It can be deducted that this node will be in permanent offline state after 5 times of poisoning.

## 4. Conclusions

In a conclusion, the core of the proposed algorithm is as following:

1) Consider the historical poisoning records of each node, which suggests whether this node is in a vulnerable condition and needs extra attention.

2) Considering the responsibility sharing mechanism of one poisoning event. It is a necessary method to rapidly evaluate the node risk. The corresponding mirror interval and poisoning grade evaluation are established, which uses storage to exchange calculation time.

3) Considering the efficiency and comprehensiveness of in-depth inspection. It starts from the data chain of original data and only runs in the simulation environment. This kind of inspection can eliminate the interference of the poisoning of the client node itself. It also has the most comprehensive data for inspection, offering a high detection rate.

4) Consider the treatment method of poisoned nodes. Use the geometric progression of offline time, which requires the administrator of the client node dealing with the working environment problems more seriously. As long as there are several problems, the corresponding node will become infinite offline.

The essence of the proposed algorithm is basically based on the general law of poisoning and uses an effective collaborative method in distributed fields such as space for time and punishment increase. The shortcoming of it is that if poisoning attack give a comprehensive data change, it is possible to avoid inspection module of simulation calculation. But for such a big scale of data modification, the design of the workload and the complexity of the poisoning program will also increase rapidly. In addition, the design of credit probability will work to trigger all the killing when the whole node poisoning risk rising steadily.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References


[1] Chaib-Draa, B. (1995) Industrial Applications of Distributed AI. *Communications of the ACM*, **38**, 49-53. https://doi.org/10.1145/219717.219761

[2] Zhang, X., Zhu, X. and Lessard, L. (2020) Online Data Poisoning Attacks. In *Learning for Dynamics and Control*, PMLR, 201-210.

[3] Jagielski, M., Severi, G., Harger, N.P. and Oprea, A. (2020) Subpopulation Data Poisoning Attacks. arXiv preprint arXiv:2006.14026 https://doi.org/10.1145/3460120.3485368

[4] Lin, F.Y.S., Wang, Y.S. and Huang, M.Y. (2013) Effective Proactive and Reactive Defense Strategies against Malicious Attacks in a Virtualized Honeynet. *Journal of Applied Mathematics*. https://doi.org/10.1155/2013/518213

[5] Cho, J.H., Sharma, D.P., Alavizadeh, H., Yoon, S., Ben-Asher, N., Moore, T.J. and Nelson, F.F. (2020) Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense. *IEEE Communications Surveys & Tutorials*, **22**, 709-745. https://doi.org/10.1109/COMST.2019.2963791

[6] Tahmasebian, F., Xiong, L., Sotoodeh, M. and Sunderam, V. (2020) Crowdsourcing under Data Poisoning Attacks: A Comparative Study. In *IFIP Annual Conference on Data and Applications Security and Privacy*, Springer, Cham, 310-332. https://doi.org/10.1007/978-3-030-49669-2_18

[7] Huang, H., Mu, J., Gong, N.Z., Li, Q., Liu, B. and Xu, M. (2021) Data Poisoning Attacks to Deep Learning Based Recommender Systems. arXiv preprint arXiv:2101.02644 https://doi.org/10.14722/ndss.2021.24525

[8] Li, R. and Asaeda, H. (2018) Secure In-Network Big Data Provision with Suspension Chain Model. *IEEE INFOCOM* 2018-*IEEE Conference on Computer Communications Workshops* (*INFOCOM WKSHPS*), 825-830. https://doi.org/10.1109/INFCOMW.2018.8406829

[9] Yuchen, S., Dejin, T., Xiaoming, Z. and Yuanchen, S. (2019) Research on Remote Sensing Image Data Attack Method Based on Machine Deep Learning Network. *Proceedings of the* 2019 2*nd International Conference on Algorithms, Computing and Artificial Intelligence*, 439-443. https://doi.org/10.1145/3377713.3377787

[10] Steinhardt, J., Koh, P.W. and Liang, P. (2017) Certified Defenses for Data Poisoning Attacks. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, 3520-3532.