Data Poison Detection Schemes for Distributed Machine Learning

YIJIN CHEN\(^1\), YUMING MAO\(^1\), HAOYANG LIANG\(^1\), SHUI YU\(^2\), (Senior Member, IEEE), YUNKAI WEI\(^1\), AND SUPENG LENG\(^1\)

\(^1\)School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
\(^2\)School of Software, University of Technology Sydney, Sydney, NSW 2007, Australia

Corresponding author: Yunkai Wei (ykwei@uestc.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0117500, in part by the EU H2020 Project COSAFE under Grant MSCA-RISE-2018-824019, in part by the Science and Technology Department of Sichuan Province under Grant 18ZDYF0329, and in part by the Joint Fund of the Ministry of Education of China and China Mobile under Grant MCM 20160304.

ABSTRACT Distributed machine learning (DML) can realize massive dataset training when no single node can work out the accurate results within an acceptable time. However, this will inevitably expose more potential targets to attackers compared with the non-distributed environment. In this paper, we classify DML into basic-DML and semi-DML. In basic-DML, the center server dispatches learning tasks to distributed machines and aggregates their learning results. While in semi-DML, the center server further devotes resources into dataset learning in addition to its duty in basic-DML. We firstly put forward a novel data poison detection scheme for basic-DML, which utilizes a cross-learning mechanism to find out the poisoned data. We prove that the proposed cross-learning mechanism would generate training loops, based on which a mathematical model is established to find the optimal number of training loops. Then, for semi-DML, we present an improved data poison detection scheme to provide better learning protection with the aid of the central resource. To efficiently utilize the system resources, an optimal resource allocation approach is developed. Simulation results show that the proposed scheme can significantly improve the accuracy of the final model by up to 20\% for support vector machine and 60\% for logistic regression in the basic-DML scenario. Moreover, in the semi-DML scenario, the improved data poison detection scheme with optimal resource allocation can decrease the wasted resources for 20-100\%.

INDEX TERMS Distributed machine learning, data poison detection, resource allocation.

I. 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 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 Lowd and Meek [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 machine learning context. Recently, there are
a 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. 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.

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 in Section VII.

II. SYSTEM MODEL

In this paper, we consider a DML system consisting of one center with a large volume training dataset $D$ at its disposal, and $T$ distributed workers $e_1, e_2, \ldots, e_T$ participating in learning from the dataset. The DML system can be basic-DML or semi-DML, three kinds of datasets are included in this paper, and a threat model is introduced to show how the attackers influence the system.

A. BASIC-DML AND SEMI-DML

In this paper, 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.

![FIGURE 1. Basic-DML vs semi-DML](image-url)

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

On the contrary, in the semi-DML scenario, the center has some spare resources in the computing server for sub-datasets learning. Consequently, it will keep some sub-datasets and learn from them by itself. That is to say, in the semi-DML, the center will learn from some sub-datasets as well as integrate the results from both of the center and distributed workers.

B. COMPONENTS OF THE SYSTEM

Due to lack of computing resource, the center will divide the training dataset $D$ into $T$ sub-datasets, i.e., $D_1, D_2, \ldots, D_T$. Bootstrap [26], one of the existing sophisticated methods for dataset dividing is adopted, to keep the statistic feature distribution of the sub-datasets $\{D_i| i \in \{1, \ldots, T\}\}$ with the training dataset $D$.

In basic-DML, the sub-datasets are assigned to the $T$ distributed workers according to the cross-learning mechanism (discussed in Section III.B). A worker $e_i (i \in \{1, \ldots, T\})$ will learn from the sub-dataset $D_m$ ($m \in \{1, \ldots, T\}$) it received, output corresponding results $w_{mi}$, and return $w_{mi}$ to the center for aggregating. Whereas in semi-DML, the sub-datasets are assigned not only to the workers but also to the center server itself, which will also produce the learning results for aggregation.

Besides the training dataset $D$ and the sub-datasets $\{D_i| i \in \{1, \ldots, T\}\}$, we define a special dataset for generating the threshold (shorten as the dataset for threshold) $D_s \subset D$ with Algorithm 1. $D_s$ has similar characters with training dataset $D$, and will be used to produce the threshold in Algorithm 1.

Fig. 2 shows the relationship of the training dataset $D$, the sub-datasets $\{D_i| i \in \{1, \ldots, T\}\}$, and the dataset for threshold $D_s$.

Besides, we use $a$ to denote the resource consumption of training a sub-dataset in the center and $b$ for that in a distributed worker. The communication resource consumed to send a sub-dataset from the center to a worker is denoted by $c$. 

VOLUME 8, 2020
For the convenience of reading, we list the main notations in Table 1.

### C. THREAT MODEL

When the attacker plans to manipulate the training result of a machine learning model, it will tamper the dataset in a well-designed way. The attacker elaborately poisons dataset with the minimal changes that make the training result much different. An attacker may have different levels of knowledge of the targeted system, such as the training data, the feature set, the learning algorithm, etc. Therefore, all the knowledge can be treated as a space $\Theta$ and the attacker’s knowledge is a subset of it. For an attacker with knowledge $\theta \in \Theta$ and a set of attacked samples $\Gamma(D_c)$, its goal can be expressed as an objective function $A(D'_c, \theta)$, where $D'_c \in \Gamma(D_c)$. The objective function shows the effect of the attack, and the goal of the attacker is to find the optimal attacked samples $D_{c*}$, which achieve the maximum attack effect:

$$D_{c*} \in \arg\max_{D'_c \in \Gamma(D_c)} A(D'_c, \theta).$$

With the above optimal attacking scheme, the attacker will try to compromise as many as possible workers into producing wrong results by tampering their assigned sub-datasets. Once a sub-dataset $D_i$ is tampered, it is turned into a poisoned sub-dataset $D'_i$. In this sense, if we have $D_i \equiv D_j \ (i, j \in \{1, \ldots, T\})$, there must be $D'_i \equiv D'_j$. This will influence our proposed scheme, we will discuss this in Section IV.

### III. DATA POISON DETECTION SCHEME IN BASIC-DML

In this section, we will discuss the data poison detection scheme in the basic-DML scenario, where the center has no spare computing resource to share for sub-dataset training tasks. In this scenario, the center only integrates the training results from the distributed workers. The data poison detection scheme in the basic-DML scenario includes three elements: a threshold of parameters, a cross-learning mechanism, and a detection method of abnormal training results.

#### A. THRESHOLD OF PARAMETERS

Many detailed internal mechanisms and principles are still left unknown in the field of machine learning [27], therefore the differences between the learned models cannot be quantified by a specific value. However, an efficient machine learning algorithm should have a good characteristic of convergence. This means if several models are learned from a dataset with the same learning algorithm, the learned models should not have significant differences.

The empirical threshold or manually set threshold is used to solve similar problems [28]–[30]. Inspired by this, in this paper we use a threshold of parameters to find out the poisoned dataset in the basic-DML scenario. We can use a threshold to distinguish the abnormal models and then find out the corresponding poisoned sub-datasets. Since the learned model consists of some parameters, we called this threshold as threshold of parameters.

To get this threshold, Algorithm 1 is proposed. This algorithm firstly selects a dataset for threshold which has the same sample distribution as the training dataset. That is to say, the dataset for threshold and the training dataset should have similar characteristics, and therefore the range of the training results from dataset for threshold can be used to find out the abnormal training results from the training dataset. The dataset for threshold is learned in the center for $t$ times to get $t$ training results, these results are sets of parameters. Finally, these groups will be used to get the threshold of parameters, which can be used for distinguishing abnormal training results from the training dataset.
Suppose the training dataset has $M$ samples. In this algorithm, we first divide the training dataset into $\varphi \eta$ parts; each part has $\frac{M}{\varphi \eta}$ samples. We then select a dataset for threshold with $\frac{M}{\varphi \eta}$ samples by $\varphi$ times, at each time we select one part and put them together to get the dataset for threshold. We use $I = I - \tau$ in this algorithm to avoid select a part repeatedly. The dataset for threshold is learned for $t$ times and generates $t$ groups of parameters (different groups of parameters will not be identical since the algorithm may not converge at the same point). Then we use the Euclidean distance between the maximum and minimum value of the parameters to get the threshold of parameters $\epsilon$.

**B. CROSS-LEARNING MECHANISM AND TRAINING LOOPS**

The cross-learning mechanism makes backups for sub-datasets to provide a foundation for finding out poisoned sub-datasets, as shown in Fig. 3. It has been demonstrated in section II that there are $T$ workers, and the data center would divide the training dataset into $T$ ($T \in N$) sub-datasets. In this mechanism, each sub-dataset will be assigned to two workers and generate two corresponding training results. For example, a sub-datasets $D_i$ is assigned to workers $e_a$ and $e_b$ ($a, b \in \{1, \ldots, T\}$). The two workers will generate two training results $w_{ia}$ and $w_{ib}$, both of which correspond to $D_i$. Therefore, there will be two training results that correspond to each sub-dataset. The algorithm of the cross-learning mechanism is shown in Algorithm 2.

If two workers received the same sub-dataset, we suppose there is a virtual connection between them. Since the sub-datasets are randomly assigned to different workers, workers may have different virtual connections according to the assigned results. In order to abstract these connections between workers, we introduce virtual topology in this part. In a virtual topology, there will be a link between two workers if they have a virtual connection (receive the same sub-dataset). Let $L = \{l_{i,j} \mid i, j \in \{1, \ldots, T\}\}$ denote the set of all links in the virtual topology. If $e_i$ and $e_j$ received the same sub-dataset, there is a link $l_{i,j}$ between them. Fig. 4 shows an example of the virtual topology. There are three loops in this figure, and we call them training loops in this paper. Based on the concept of virtual topology and training loops, we can get a lemma as below.
Algorithm 3 Data poison detection scheme.

Input:
- Set of workers $E = \{e_i | i \in \{1, \ldots, T\}\}$;
- Set of training results (parameter of model) from sub-dataset $m$: $p_m = \{w_{m,1}, w_{m,2}\}$;
- All divided training dataset $\{D_m | m \in \{1, \ldots, T\}\}$;
- Poisoned sub-dataset set: $P = \emptyset$

% Compare two results from the same sub-dataset, and put the poisoned sub-dataset into $P$
1: for $m = 1 : T$ do
2: if $||w_{m,1} - w_{m,2}||_2 \geq \epsilon$ then
3: Add $D_m$ to $P$
4: end if
5: end for

Output:
- Set of poisoned sub-datasets: $P$

between these two groups to measure their difference, and denote it as $||w_{1,e} - w_{1,f}||_2$. If the difference between the two groups of parameters sets trained from the same sub-dataset is smaller than $\epsilon$, the sub-dataset is considered to be unpoisoned. Hence the result of this sub-dataset can be used to update the parameters of the trained model in the center. On the contrary, if the difference is bigger than $\epsilon$, the sub-dataset is considered to be poisoned. Thus the center would resend this sub-dataset for relearning. This scheme is shown in Algorithm 3.

However, the proposed scheme in the basic-DML scenario is not perfect and cannot find out all the poisoned sub-datasets. For example, as discussed in Section III, if the attacker compromises two workers with the same sub-dataset, this poisoned sub-dataset cannot be found since the center cannot differentiate them with the proposed scheme. Moreover, even though this scheme can find out the poisoned sub-dataset, it cannot identify which of the two corresponding workers is compromised. This deficiency of the data poison detection scheme in the basic-DML scenario is demonstrated in Fig. 5.

### D. PROBABILITY OF FINDING THREATS

#### IN DIFFERENT SITUATIONS

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
Algorithm 4. With the help of central resources, the improved data poison detection scheme increases if there are more training loops. Therefore, the validity of the proposed data poison detection scheme increases if there are more training loops.

The number of workers can influence the difficulty for attackers to poison the dataset. In a network, the dataset is distributed evenly to each worker. Compared with a network with fewer workers, it is more difficult for attackers to poison the same amount of dataset in a network with more workers since they need to compromise more. Therefore, the validity of the proposed data poison detection scheme increases when the number of workers increases.

The statistic characteristics of the special dataset will influence the threshold of parameters. This threshold is used for detecting the abnormal results from the sub-dataset and find the poisoned sub-dataset. Therefore, if there are more similarities of the statistic characteristic between the special dataset and the sub-datasets, the validity of the proposed data poison detection scheme will be higher.

IV. DATA POISON DETECTION SCHEME IN SEMI-DML

In this section, we discuss the improved data poison detection scheme (hereinafter referred to as the improved scheme) in the semi-DML scenario, where the center shares spare resources in the dataset training tasks. Based on the three elements of the data poison detection scheme in the DML scenario, the improved scheme includes one more element: central assistance. In this scenario, the center can learn part of all the sub-datasets, or verify the results from workers by relearning the suspicious sub-datasets. With the assistance of the center, the resource cost of the system will vary with different resource allocation schemes of the central resources. Therefore how to make efficient utilization of the system resources is the essential problem in this scenario.

As shown in Fig. 5, although the data poison detection scheme in the DML scenario can find out the poisoned sub-dataset, it cannot distinguish which of the two corresponding workers is compromised. To solve this problem, we present an improved data poison detection scheme with center resources aided in the semi-DML. The improved scheme is shown in Fig. 6 and the algorithm is described in Algorithm 4. With the help of central resources, the improved scheme can identify the abnormal one in two suspicious results and hence distinguish the corresponding compromised worker, which can not be realized in the DML scenario.

E. THREATS TO VALIDITY IN BASIC-DML

In basic-DML, there are several elements can influence the validity of the proposed scheme: the number of training loops, the number of workers, and the statistic characteristic of the dataset for threshold.

The number of training loops affects the validity by the PFT. In a 1-loop situation, all the workers are adjacent to each other while in a multi-loop situation, two workers are non-adjacent if they are not on the same loop. When two adjacent workers are compromised simultaneously, the poisoned dataset can not be detected, which can more likely happen in a 1-loop situation. Therefore, the validity of the proposed data poison detection scheme increases if there are more training loops.

The number of workers can influence the difficulty for attackers to poison the dataset. In a network, the dataset is distributed evenly to each worker. Compared with a network with fewer workers, it is more difficult for attackers to poison the same amount of dataset in a network with more workers since they need to compromise more. Therefore, the validity of the proposed data poison detection scheme increases when the number of workers increases.

The statistic characteristics of the special dataset will influence the threshold of parameters. This threshold is used for detecting the abnormal results from the sub-dataset and find the poisoned sub-dataset. Therefore, if there are more similarities of the statistic characteristic between the special dataset and the sub-datasets, the validity of the proposed data poison detection scheme will be higher.
In this scheme, the center can improve the security of distributed training with spare central resources. In this case, two actions can be conducted with the central resources: (1) Learning a part of the sub-datasets directly to ensure its accuracy. (2) Verifying two suspicious sub-datasets to find out the poisoned sub-dataset and the compromised worker. Both of the two actions can improve the security of the given system when the poison attack occurs. We can use part of the central resources for learning and the rest for verification.

However, there exists a waste of resources in some situations: (1) If the cost of the resources for verifying sub-datasets is not adequate, some suspicious sub-datasets have to be trained in the workers again, which will cause the resources wasted of the workers. (2) If the cost of the resources for verifying sub-dataset are excessive, this extra verification resource will be wasted.

To minimize the wasted resource, we intend to find the optimal allocation of the central resources for learning and verification. Furthermore, we will compare the optimal allocation scheme with the learning-only scheme and verification-only scheme, where the central resources are completely denoted in learning and verification.

### A. OPTIMAL ALLOCATION SCHEME FOR THE SEMI-DML SCENARIO

The wasted resource $W$ in the system is related to three parameters: $p$, $\alpha$, and $R$, where $p$ is the compromised probability of a distributed worker; $\alpha$ is the proportion of training resources in total center resources; and $R$ is the amount of center resources. The waste of the system can be computed as follow:

$$W(p, \alpha, R) = \begin{cases} (1 - \alpha)R - n(p, K)a, & (1 - \alpha)R > n(p, K)a \\ (n(p, K) - \frac{(1-\alpha)R}{\alpha})b, & (1 - \alpha)R < n(p, K)a \\ 0, & (1 - \alpha)R = n(p, K)a, \end{cases}$$

(8)

where $K$ is the number of sub-datasets learned on the distributed workers, and $n(p, K)$ is the number of compromised workers when there are $N_w$ workers in total and the compromised probability of each worker is $p$. $K$ and $n(p, N_w)$ are calculated as follows:

$$K = T - \lfloor \frac{\alpha R}{a} \rfloor,$$

(9)

$$n(p, N_w) = \sum_{i=0}^{N_w} iC_i^{N_w}p^i(1-p)^{N_w-i}.$$  

(10)

The optimal allocation scheme aims to get the value of $\alpha$, which can minimize the wasted resources in the proposed scheme. Therefore we use an optimization function to find the suitable value of $\alpha$ for the minimum wasted resources as below:

$$\min W(p, \alpha, R)$$

s.t. $0 \leq \alpha \leq 1.$  

(11)

After solving the above function, we can get the optimal value of $\alpha$. We use $\alpha'$ to denote the optimal solution of $\alpha$ and analyze the waste rate and correct rate for the optimal scheme later.

Before computing the waste rate, we need to get the total resource consumption of the optimal resource allocation scheme, which is denoted as $R_{total}$. The total resource consumption of the optimal allocation scheme is the sum of all the resources in the center and the consumptive resources on the distributed workers:

$$R_{total} = R + (T - \lfloor \frac{\alpha' R}{a} \rfloor)b.$$

(12)

Based on the total resource consumption and the wasted resources, we can get the waste rate of the optimal allocation scheme $wr$ as follow:

$$wr = \frac{W(p, \alpha', R)}{R_{total}}.$$  

(13)
When the verification resources in the center is enough to verify all the suspicious sub-datasets, the correct rate of the optimal allocation scheme is 100%. Otherwise, the correct rate is related to the number of suspicious sub-datasets and the resources used for verification. Therefore the correct rate of the optimal allocation scheme \( cr \) is computed as follow:

\[
cr(\alpha, R, p) = \begin{cases} 
1, & (1 - \alpha)R \geq n(p, K)a \\
1 - \frac{n(p, K) - \lfloor \frac{(1 - \alpha) R}{T} \rfloor}{T}, & (1 - \alpha)R < n(p, K)a.
\end{cases}
\] (14)

**B. LEARNING-ONLY SCHEME FOR THE SEMI-DML SCENARIO**

The learning-only scheme means reserving all the center resources for learning the sub-datasets directly, where \( \alpha = 1 \). In this part, we will discuss how to compute the wasted resources, waste rate, and correct rate of the learning-only scheme.

The wasted resources of the learning-only scheme \( W_1 \) are the resources which are used to retraining suspicious sub-datasets in distributed workers:

\[
W_1(p, R) = n(p, T - \lfloor \frac{R}{a} \rfloor)b.
\] (15)

The waste rate of the learning-only scheme \( w_{r1} \) is the proportion of wasted resources in the whole cost resources:

\[
w_{r1} = \frac{W_1(p, R)}{R + (T - \lfloor \frac{R}{a} \rfloor)b}.
\] (16)

The correct rate \( cr_1 \) is related to the number of suspicious sub-datasets and computed as follow:

\[
cr_1 = 1 - \frac{n(p, T - \lfloor \frac{R}{a} \rfloor)}{T},
\] (17)

where \( \frac{n(p, T - \lfloor \frac{R}{a} \rfloor)}{T} \) is the error rate in this scheme.

**C. VERIFICATION-ONLY SCHEME FOR THE SEMI-DML SCENARIO**

The verification-only scheme means using all the center resources for verifying suspicious sub-datasets, where \( \alpha = 0 \). When the resources in the center are excessive to verify all the suspicious datasets, the wasted resources are the extra resources in the center. On the contrary, if the resources in the center are not enough to finish all the verification tasks, it will waste resources of distributed workers to retrain suspicious datasets. The wasted resources \( W_2 \) is computed as follow:

\[
W_2(p, R) = \begin{cases} 
R - n(p, T)a, & R > n(p, T)a \\
n(p, T) - \frac{R}{a}b, & R \leq n(p, T)a.
\end{cases}
\] (18)

The whole cost resources in this scheme are all the resources in the center and the training resources in the distributed workers. Therefore the waste rate \( w_{r2} \) is calculated as:

\[
w_{r2} = \frac{W_2(p, R)}{R + Tb}.
\] (19)

In the verification-only scheme, when the central resources are excessive to verify all the suspicious sub-datasets, the correct rate is 100%, otherwise it is less than 100% and related to the number of suspicious sub-datasets. Therefore, the correct rate is computed as:

\[
cr_2 = \begin{cases} 
1, & R \geq n(p, T)a \\
1 - \frac{n(p, T) - \lfloor \frac{R}{a} \rfloor}{T}, & R < n(p, T)a.
\end{cases}
\] (20)

**D. THREATS TO VALIDITY IN SEMI-DML**

The difference between the semi-DML and the basic-DML is only related to the center, so the workers and datasets in these two scenarios have the same characteristic. Since all the threats to validity in basic-DML are about the worker and the dataset, the threats in the basic-DML are also threats in the semi-DML.

Moreover, in the semi-DML scenario, the allocation of resources between learning and verification can influence the validity of the proposed scheme. The efficiency of allocation depends on the number of suspicious sub-datasets. To guarantee the validity, the center should reserve enough resource for verification of the suspicious dataset. However, a simple increase of reserved resources for verification does not mean a better resource allocation due to the waste. Therefore the optimal resource allocation is necessary.

**V. PERFORMANCE EVALUATION**

In this section, we conduct simulations for the previously proposed data poison detection scheme in both DML and semi-DML scenarios to evaluate their performance. The simulations are conducted on the Python platform and Wolfram Mathematica and the details of the simulations in two scenarios are described below. We conduct the simulation based on the SVM algorithm, which is one of the most classical algorithms of machine learning and suitable for the classification of the massive dataset with high-dimension data in the DML. Moreover, to verify the performance of the proposed scheme on different algorithms, we conduct one more simulation based on the LR.

**A. SIMULATION IN THE BASIC-DML SCENARIO**

We firstly use multi-process to simulate the distributed system on the Python platform and implement the proposed data poison detection scheme on the system. Then, we use the support vector machine (SVM) algorithm to learn a dataset that is generated by the machine learning library called scikit-learn. The trained model by SVM is compared with the mathematical results conducted by another platform called Wolfram Mathematica. Moreover, we use another Logistic Regression (LR) algorithm and compare its results with that in SVM to evaluate the performance of the proposed scheme with different learning algorithms. The dataset used for LR is from [31], and all the parameters of the simulation are listed in Table 2. According to Eq. (1), the attacker will always use the optimal attacked strategy for attacking. Therefore, in the
TABLE 2. Parameters of simulation.

|        | samples | features | workers | running times |
|--------|---------|----------|---------|---------------|
| SVM    | 10000   | 20       | 20      | 100           |
| LR     | 42000   | 784      | 20      | 100           |

Firstly, we intend to validate the proposed mathematical model by comparing the model results with the simulation results. The model results are got from our proposed mathematical model, which is concerned in Section III-D. The comparison between the model results and the simulation results is shown in Fig. 7. From this figure, we can see that the results of the mathematical model match the simulation results well, which indicates the proposed mathematical model can accurately obtain the PFT in the proposed scheme. Furthermore, both of the results clearly show that the optimal number of training loops is the maximum of \( k \), which is 10 in the simulation.

Fig. 8 shows the PFT in the basic-DML scenario when \( k = 1, 4, 7, 8, 9, 10 \). We can see from the figure that the three lines of \( k = 1, 4, 7 \) are very close to each other and have a similar tendency, which decreases rapidly when \( y \) increases from 1 to 6 and then decreases slowly to 0 when \( y \) increases from 6 to 10. The PFT has an evident increase from the line of \( k = 7 \) to the line of \( k = 8 \) and keeps increasing until it reaches the maximum value when \( k = 10 \).

Fig. 9 demonstrates the classification accuracy of the basic-DML with SVM in three situations: without data poisoning, data poisoning without the proposed scheme, and data poisoning with the proposed scheme. In the basic-DML scenario without data poisoning, the classification accuracy is near to 93%. If the basic-DML system is influenced by data poisoning, the classification accuracy would gradually decrease from 93% to nearly 50% with the number of compromised workers increases. Nevertheless, the proposed data poison detection scheme can increase the classification accuracy in the basic-DML scenario with data poisoning. When half of the workers are compromised, the proposed scheme can keep the classification accuracy near to 84%, which is 20% higher than the case without the proposed scheme.

Compared with the SVM algorithm in Fig. 9, the classification accuracy of basic-DML with LR in the same three situations is demonstrated in Fig. 10. Since the SVM algorithm is more sensitive to abnormal data than the LR algorithm, the classification accuracy of SVM already decreases with 3 compromised workers while the classification accuracy of LR decreases with 7 compromised workers. Therefore, the classification accuracy of the SVM decreases earlier than the LR algorithm. These two figures show that the proposed data poison detection scheme has a good effect even on data sensitive algorithm.

B. SIMULATION IN THE SEMI-DML SCENARIO

We use the Wolfram Mathematica to conduct the numerical simulation of the wasted resources, waste rate, and correct rate in the semi-DML scenario. In the simulation, we compare...
the performance of three schemes: the optimal allocation scheme, learning-only scheme, and verification-only scheme.

Fig. 11 shows the wasted resources of these three schemes with different center resources $R$. The optimal resources allocation scheme has minimal wasted resources in these three schemes. Moreover, with the increasing of infected probability $p$, the wasted resources of the optimal scheme are gradually close to that of the verification-only scheme, and finally they close to each other when $p$ is big enough. It also shows that with the increase of the infected probability, the wasted resources of the optimal allocation scheme and learning-only scheme are always increasing. However, the wasted resources of the verification-only scheme are gradually decreasing at the beginning and then increase. Furthermore, with the increase of center resources, the advantage of the optimal allocation scheme becomes more evident. When the amount of center resources is sufficient, the optimal allocation scheme can keep the wasted resource nearly to 0.

Fig. 12 shows the waste rate of the three schemes. From this figure, the waste rate of the optimal allocation scheme is always less than the other two schemes. Comparing this figure with Fig. 11, we can see that even though the wasted resources of the verification-only scheme are sometimes equal to the optimal allocation scheme, the resource allocation scheme always has a lower waste rate than the verification-only scheme. Besides, Fig. 13 shows the correct

rate of the three schemes. In this figure, the optimal allocation scheme and the verification-only scheme has the highest correct rate, and the learning-only scheme has a lower one.

From the above three figures, the simulation results show that the optimal allocation scheme has a better performance than the other two schemes: it has a minimal waste rate as well as the highest correct rate. The verification-only scheme has the same correct rate as the optimal allocation scheme. Nevertheless, the verification-only scheme has the highest waste rate when the center resources are sufficient enough or when the infected probability is low enough. The learning-only scheme seems to be the worst scheme since it does not have an advantage over the other two schemes.

VI. SUMMARY AND FUTURE WORK

In this paper, we discussed the data poison detection schemes in both basic-DML and semi-DML scenarios. The data poison detection scheme in the basic-DML scenario utilizes a threshold of parameters to find out the poisoned sub-datasets. Moreover, we established a mathematical model to analyze the probability of finding threats with different numbers of training loops. Furthermore, we presented an improved data poison detection scheme and the optimal resource allocation in the semi-DML scenario. Simulation results show that in the basic-DML scenario, the proposed scheme can increase the model accuracy by up to 20% for support vector machine and
60% for logistic regression, respectively. As to the semi-DML scenario, the improved data poison detection scheme with optimal resource allocation can decrease wasted resources for 20-100% compared to the other two schemes without the optimal resource allocation.

In the future, the data poison detection scheme can be extended to a more dynamic pattern to fit the changing application environment and attacking intensity. Besides, since the multi-training of sub-datasets would increase the resource consumption of the system, the trade-off between security and resource cost is another topic that needs to be studied further.

APPENDIXES

APPENDIX A

Let \( \{D_m|m \in \{1,\ldots,T\}\} \) denote the set of all sub-datasets. Supposing that the sub-dataset \( D_{\xi 1} \) is sent to workers \( n_1 \) and \( n_2 \), we can get \( L = \{l_{<n_1,n_2>, l_{<n_2,n_1>}\} \). Then another sub-dataset \( D_{\xi 2} \) is sent to workers \( n_2 \) and \( n_3 \), so we get \( L = \{l_{<n_1,n_2>, l_{<n_2,n_3>}, \ldots, l_{<n_T,n_1>}\} \). In the next round, there are two possible situations:

1. The sub-dataset \( D_{\xi (\xi+1)} \) is sent to workers \( n_{\xi+1} \) and \( n_1 \). In this situation, we can get \( L = \{l_{<n_1,n_2>, l_{<n_2,n_3>}, \ldots, l_{<n_{\xi+2},n_{\xi+1}>}, l_{<n_{\xi+1},n_{\xi+2}>}\} \) and all the links in \( L \) will form the first training loop. After that, new links would appear and generate other training loops with the similar process of the first loop.
2. The sub-dataset \( D_{\xi (\xi+1)} \) is sent to worker \( n_{\xi+1} \) but not sent to worker \( n_1 \). This situation will return to 1 if later a sub-dataset is sent to worker \( n_1 \). On the contrary, if no any sub-dataset but the last one is sent to worker \( n_1 \), there would be just one training loop in the virtual topology.

APPENDIX B

The proposed scheme can find the threat only when the attacker has not compromised adjacent workers. Based on the permutation and combination theory, if the attacker has randomly compromised \( y \) workers in a loop with \( x \) workers, there are two situations where all the \( y \) workers are non-adjacent, as shown in Fig. 14: (a) The attack has compromised \( y \) non-adjacent workers of \( x - 1 \) workers in a line and there are \( C_{x-1}^y \) possible cases; (b) The attack has compromised \( y - 1 \) non-adjacent workers of \( x - 3 \) workers in a line and there are \( C_{x-y-1}^{y-1} \) cases. So when the attacker has randomly compromised \( y \) workers in a loop with \( x \) workers, there are totally \( C_{x-y-1}^{y-1} + C_{x-1}^y \) cases in which we can find the threat. Note that if \( y > \frac{x}{2} \), there must be some workers adjacent to each other. So the PFT in a 1-loop situation is as follow:

\[
H(x, y) = \begin{cases} 
0 & y > \frac{x}{2} \\
C_{x-y-1}^{y-1} + C_{x-1}^y & 0 < y \leq \frac{x}{2}
\end{cases}
\]  
(21)

APPENDIX C

In a \( k \)-loop situation, we suppose the \( i \)-th loop has \( x_i \) workers (\( i \in \{1, k\} \) workers). The combinations of \( x_1, \ldots, x_k \) satisfy the following equation:

\[
x_1 + x_2 + \ldots + x_k = x.
\]  
(22)

Suppose there are totally \( y \) compromised workers in these \( k \) loops, and the attacker has compromised \( y_i \) workers in the \( i \)-th loop (\( i \in \{1, k\} \) loop). The PFT of this case is:

\[
P_k^{x_1,\ldots,x_k}(y_1, \ldots, y_k) = H(x_1, y_1)H(x_2, y_2)\ldots H(x_k, y_k),
\]  
(23)

where \( y_1, y_2, \ldots, y_k \) have many possible values satisfying the following equation:

\[
y_1 + y_2 + \ldots + y_k = y.
\]  
(24)

and the occurring probability of each situation is:

\[
\xi_k^{x_1,\ldots,x_k} = \frac{C_{x_1}^y \ldots C_{x_k}^y}{C_x^y}.
\]  
(25)

We use \( P_k^{x_1,\ldots,x_k}(y) \) to denote the mathematical expectation of \( P_k^{x_1,\ldots,x_k}(y_1, \ldots, y_k) \), it means the PFT when there are \( y \) compromised workers in \( k \) loops, where the \( i \)-th loop has \( x_i \) workers (\( i \in \{1, k\} \) workers). It is computed as follows:

\[
P_k^{x_1,\ldots,x_k}(y) = E[P_k^{x_1,\ldots,x_k}(y_1, \ldots, y_k)]
= \sum_{y_1=\psi_1^k}^{\psi_2^k} \ldots \sum_{y_k=\psi_1^k}^{\psi_2^k} \xi_k^{x_1,\ldots,x_k}(y_1, \ldots, y_k).
\]  
(26)

where \( \psi_1^m \) and \( \psi_2^m \) are the lower bound and upper bound of \( y_{m(m \in \{1, \ldots, k\})} \), respectively. They are expressed as follow:

\[
\psi_1^m = \max[y - (y_1 + \ldots + y_{m-1}) - (x_{m+1} + \ldots + x_k), 0],
\]  
(27)

\[
\psi_2^m = \min[y - (y_1 + \ldots + y_{m-1}), x_2].
\]  
(28)
APPENDIX D

Based on Appendix VI, $x_1, x_2, \ldots, x_k$ have many possible values which satisfy Eq. (22) in $P^k_{\{x_1, \ldots, x_k\}(y)}$. We suppose $\delta^k(x_1, \ldots, x_k)$ is the occurring probability of a $k$-loop situation, where the $i$-th ($i \in [1, k]$) loop has $x_i$ workers. Therefore, the mathematical expectation of $P^k_{\{x_1, \ldots, x_k\}(y)}$ is:

$$E^k(x, y) = E[P^k_{\{x_1, \ldots, x_k\}(y)}]$$

$$= \sum_{x_1=2}^{\psi^k_E} \sum_{x_2=1}^{\psi^k_E} \cdots \sum_{x_k=1}^{\psi^k_E} \delta^k(x_1, \ldots, x_k) P^k_{\{x_1, \ldots, x_k\}(y)},$$

(29)

where the occurring probability $\delta^k(x_1, \ldots, x_k)$ is computed as follow:

$$\delta^k(x_1, \ldots, x_k) = \frac{C^2_{x_1} C^2_{x_2} \cdots C^2_{x_k-1}}{k^k},$$

(30)

and $\psi^k_E$ is the upper bound of $x_m (m \in \{1, \ldots, k\})$.

$$\psi^k_E = x - (x_1+1, \ldots, x_m-1) - 2(k-m).$$

(31)

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

[8] 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.

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

[10] 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.

[11] A. Alfeld, X. Zhu, and P. Bartford, “Data poisoning attacks against autoregressive models,” in Proc. 13th AAAI Conf. Artif. Intell., Feb. 2016.
HAOYANG LIANG received the B.E. degree in Internet of Things engineering from the University of Electronic Science and Technology of China, in 2013, where he is currently pursuing the master’s degree with the School of Information and Communication Engineering. His research interest is in the domain of communication and information engineering, especially relating to artificial intelligence-based communication network security and blockchain.

SHUI YU (Senior Member, IEEE) is currently a Professor with the School of Computer Science, University of Technology Sydney, Australia. He has published two monographs and edited two books and more than 300 technical articles, including top journals and top conferences, such as the IEEE TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. His research interests include security and privacy, networking, big data, and mathematical modeling. He initiated the research field of networking for big data in 2013. His h-index is 41. He is a member of AAAS and ACM. He is currently serving a number of prestigious editorial boards, including the IEEE Communications Surveys and Tutorials (Area Editor) and the IEEE Communications Magazine. He is a Distinguished Lecturer of the IEEE Communication Society.

YUNKAI WEI received the B.Eng., M.Eng., and Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China. He is currently an Associate Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. He was a Visiting Researcher with the California Institute of Technology from 2013 to 2014. His research interests include cyber security, 5G and beyond communications, and the Internet of Things.

SUPENG LENG received the Ph.D. degree from Nanyang Technological University (NTU), Singapore. He is currently a Full Professor and the Vice Dean of the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC). He is also the Leader of the research group of Ubiquitous Wireless Networks. He has been a Research Fellow with the Network Technology Research Center, NTU. His research focuses on resource, spectrum, energy, routing, and networking in the Internet of Things, vehicular networks, broadband wireless access networks, smart grids, and the next-generation mobile networks. He has published over 180 research articles in recent years. He serves as the Organizing Committee Chair and a TPC Member for many international conferences, as well as a Reviewer for over ten international research journals.

* * *