

**DATA POISON DETECTION SCHEMES FOR DISTRIBUTED MACHINE
LEARNING****RANGU POOJITHA¹, DR. K. KISHOR KUMAR²**

¹ PG Student, Department of Computer Science and Engineering, Kakatiya University College Of Engineering and Technology Warangal-506009, (TS).

² Associate Professor, Department of Computer Science and Engineering, Kakatiya University, Warangal-506009, (TS).

¹ rangupoojitha@gmail.com, ² k_kishorkumar@yahoo.com

ABSTRACT

In situations when a single node is unable to provide correct results in a reasonable amount of time, distributed machine learning, or DML, may produce enormous dataset training. In contrast to a non-distributed system, this will unavoidably expose more possible targets to attackers. In this work, we divide DML into two categories semi-DML and basic-DML. The centre server assigns learning assignments to dispersed machines in basic-DML, then aggregates the devices' learning outcomes. In semi-DML, the central server dedicates additional resources to dataset learning, on top of its responsibilities in basic-DML. First, we propose a

unique data poisoning detection approach for basic-DML that determines the data that is poisoned by means of a cross-learning process. We demonstrate that the suggested cross-learning process would produce training loops, which serve as the foundation for the development of a mathematical model that determines the ideal number of training loops. Then, with the use of the central resource, we provide an enhanced data poisoning detection strategy for semi-DML to better safeguard learning. A technique for optimum resource allocation is created in order to make effective use of the system resources. According to simulation data, in the basic-DML scenario, the suggested

strategy may greatly increase the final model's accuracy by as much as 20% for support vector machines and 60% for logistic regression. Furthermore, the optimum resource allocation in the enhanced data poison detection technique may reduce resource waste by 20–100% in the semi-DML scenario.

Index Terms: - Distributed machine learning, data poison detection, resource allocation

1.INTRODUCTION

In distributed systems, when no one node can get an intelligent judgement from a large dataset in a reasonable amount of time, distributed machine learning, or DML, has been used extensively. A central server in a typical DML system has access to enormous amounts of data. The dataset is split up and distributed to dispersed workers who complete training tasks and send back their findings to the central location. Ultimately, the centre produces the final model by

integrating these findings. Regretfully, it is becoming more difficult to ensure each worker's security as the number of dispersed workers rises. The likelihood that attackers may tamper with the dataset and alter the training outcome will rise as a consequence of this security lapse. One common method used in machine learning to interfere with training data is the poisoning attack. An attacker will have additional opportunities to contaminate the datasets, which will increase the danger in DML, particularly in cases where freshly created datasets must be provided to the dispersed workers on a regular basis for updating the decision model. Scholars are paying close attention to this machine learning risk. Initially, Dalvi et al. showed that if an attacker had all the knowledge, they could modify the data to outwit the data miner. Then, Lowd argued that the assumption of perfect information is implausible, demonstrating that attackers may create assaults using just

a portion of the available data. Following that, a number of works were carried out, concentrating on the situation of non-distributed machine learning. There have been several recent initiatives focused on preventing data manipulation in DML. For instance, Zhang and Esposito et al. created a safe approach for collaborative deep learning and distributed support vector machine (DSVM) using game theory. These schemes, however, are not applicable to typical DML scenarios since they are created for a particular DML algorithm. It is critical to investigate a broadly applicable DML protection mechanism since an adversarial attack has the potential to mislead different machine learning algorithms.

Depending on whether the centre distributes resources in the dataset training tasks, we categorise DML in this project as basic distributed machine learning (basic-DML) and semi-distributed machine learning

(semi-DML). We then provide data poisoning detection strategies for basic and semi-DML, respectively. The effectiveness of our suggested systems is confirmed by the experimental findings.

PURPOSE:

Both Basic DML and Semi DML are distributed methodologies that we use. In Basic DML, a dataset is divided into many sections and sent to worker nodes, who then create an ML model and return the output back to the distributed centre server. To train a dataset for an ML model, the Semi DML Centre server will provide resources to it.

PROBLEM STATEMENT:

In situations when a single node is unable to provide correct results in a reasonable amount of time, distributed machine learning, or DML, may produce enormous dataset training. In contrast to a non-distributed system, this will unavoidably expose more

possible targets to attackers. In this work, we divide DML into two categories: semi-DML and basic-DML.

II. LITERATURE SURVEY

Guanhua Qiao; Supeng Leng proposed an emerged as a promising paradigm to realize user requirements with low-latency applications. The deep integration of multi-access technologies and MEC can significantly enhance the access capacity between heterogeneous devices and MEC platforms. However, the traditional MEC network architecture cannot be directly applied to the Internet of Vehicles (IoV) due to high-speed mobility and inherent characteristics. Furthermore, given a large number of resource-rich vehicles on the road, it is a new opportunity to execute task offloading and data processing onto smart vehicles. To facilitate good merging of the MEC technology in IoV, this article first introduces a

vehicular edge multi-access network that treats vehicles as edge computation resources to construct the cooperative and distributed computing architecture. For immersive applications, co-located vehicles have the inherent properties of collecting considerable identical and similar computation tasks. We propose a collaborative task offloading and output transmission mechanism to guarantee low latency as well as the application-level performance. Finally, we take 3D reconstruction as an exemplary scenario to provide insights on the design of the network framework. Numerical results demonstrate that the proposed scheme is able to reduce the perception reaction time while ensuring the application-level driving experiences. Ke Zhang; Supeng Leng described research on the Internet of Things (IoT) platform has played a significant role in improving road transport safety and efficiency by ubiquitously connecting intelligent

vehicles through wireless communications. Such an IoT paradigm however, brings in considerable strain on limited spectrum resources due to the need of continuous communication and monitoring. Cognitive radio (CR) is a potential approach to alleviate the spectrum scarcity problem through opportunistic exploitation of the underutilized spectrum. However, highly dynamic topology and time-varying spectrum states in CR-based vehicular networks introduce quite a few challenges to be addressed. Moreover, a variety of vehicular communication modes, such as vehicle-to-infrastructure and vehicle to-vehicle, as well as data QoS requirements pose critical issues on efficient transmission scheduling. Based on this motivation, in this paper, we adopt a deep Q -learning approach for designing an optimal data transmission scheduling scheme in cognitive vehicular networks to minimize transmission costs while

also fully utilizing various communication modes and resources. Furthermore, we investigate the characteristic modes and spectrum resources chosen by vehicles indifferent network states, and propose an efficient learning algorithm for obtaining the optimal scheduling strategies. Numerical results are presented to illustrate the performance of the proposed scheduling schemes. Tianqi Chen, Mu Li, and Yutian Li an implemented a MX Net. MX Net is a Multilanguage machine learning (ML)library to ease the development of ML algorithms, especially for deep neural networks. Embedded in the host language, it blends declarative symbolic expression with imperative tensor computation. It offers auto differentiation to derive gradients. MXNet is computation and memory efficient and runs on various heterogeneous systems, ranging from mobile devices to distributed GPU clusters. This paper describes both the

API design and the system implementation of MXNet, and explains how embedding of both symbolic expression and tensor operation is handled in a unified fashion. Our preliminary experiments reveal promising results on large scale deep neural network applications using multiple GPU machines. Lina Zhou, Shimei Pan and Jianwu Wang proposed the approach for machine learning model. Machine learning (ML) is continuously unleashing its power in a wide range of applications. It has been pushed to the forefront in recent years partly owing to the advent of big data. ML algorithms have never been better promised while challenged by big data. Big data enables ML algorithms to uncover more fine-grained patterns and make more timely and accurate predictions than ever before; on the other hand, it presents major challenges to ML such as model scalability and distributed computing. In this paper, we introduce a

framework of ML on big data (MLBiD) to guide the discussion of its opportunities and challenges. The framework is centered on ML which follows the phases of preprocessing, learning, and evaluation. In addition, the framework is also comprised of four other components, namely bigdata, user, domain, and system. The phase of ML and the components of MLBiD provide directions for the identification of associated opportunities and challenges and open up future work in many unexplored renders or under explored research areas. J. Chen et al. described Deep Poisons an innovative hostile network with one generator and two distinctions to solve this problem. In particular, the generator automatically extracts hidden features of the target class and embeds them in harmless training models. A discriminator controls the rate of addiction harassment. Another discriminator acts as a target model to demonstrate the effects of the drug.

The novelty of Deep Poisons that the toxic training models developed cannot be distinguished from harmless lessons by defensive methods or human visual inspection, and even harmless test models can be attacked the evolution of malware. However, it can also lead to toxic attacks, especially backdoor attacks, which disrupt the learning process and create evasion tunnels for manipulated malware models. No previous research has examined this critical issue with Android Malware Detector. J. Chen et al. described, Advanced attackers may be vulnerable to data poisoning attacks and may interfere with the learning process by inserting some malicious samples into the training database. Existing defenses against drug attacks are primarily target-specific attacks. Designed for a specific type of attack. However, due to the explicit principles of the Master, it does not work for other types. However, some common safety strategies have

C. Li et al. described, Machine Learning (ML) is widely used to detect malware on various platforms, including Android. Detection models must be retested following the data collected (e.g., monthly) to continue developed.

III. EXISTING SYSTEM

Regretfully, it is becoming more difficult to ensure each worker's security as the number of dispersed workers rises. The likelihood that attackers may tamper with the dataset and alter the training outcome will rise as a consequence of this security lapse. One common method used in machine learning to interfere with training data is the poisoning attack. An attacker will have additional opportunities to contaminate the datasets, which will increase the danger in DML, particularly in cases where freshly created datasets must be provided to the dispersed workers on a regular basis for updating the decision model.

A supervised machine learning technique called "Support Vector Machine" (SVM) may be used to problems involving both regression number of features you have) using the SVM method, with each feature's being represented by a specific coordinate. Next, we carry out the classification process by identifying the hyper-plane that effectively separates the two classes.

DISADVANTAGES OF EXISTING SYSTEM:

The results are not accurate with naive bayes and support vector machine

IV PROPOSED SYSTEM:

We use the Data Poison Detection approach to identify and eliminate altered data because in a distributed setting, attackers may alter training data and use machine learning to anticipate incorrect results. This method will go over the training dataset to find any unusual values, then eliminate them. The Data Poison

and classification. It is usually used to categorization difficulties, however. Each data item is plotted as a point in n-dimensional space (where n is the

approach allows us to increase the accuracy of machine learning systems. Semi-DML and Basic-DML As shown in the graphic below, we are using Basic DML and Semi DML, two distributed approaches, in our project. A basic DML will split the dataset into many pieces and deliver it to worker nodes, which will then create an ML model and return the finished product back to the distributed centre server. The Semi DML Centre server will provide resources to train the ML model on the given dataset.

ADVANTAGES OF PROPOSED SYSTEM:

Time Consumption is less.

More accurate results are to be seen.

V. SYSTEM DESIGN

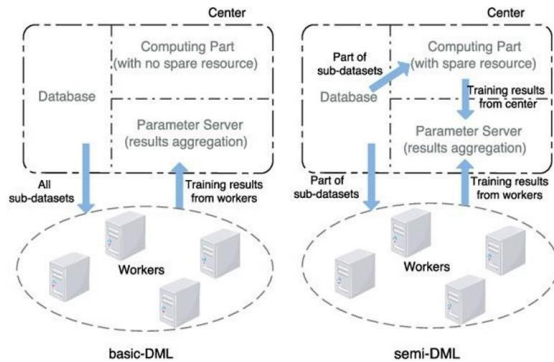


Fig1: Architecture of system.

DML is divided into two categories, basic DML and semi-DML, as shown in the above image. There is a centre in both cases, and it houses a parameter server, a compute server, and a database. But in these two cases, the centre serves distinct purposes.

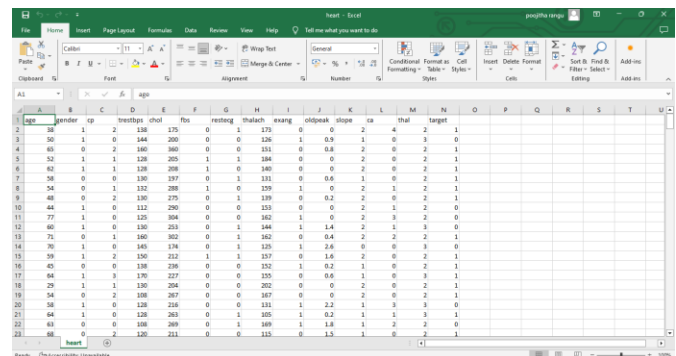
In the basic-DML scenario, all sub-datasets will be sent to the dispersed workers since the centre lacks spare computer resources for sub-dataset training. As a consequence, the parameter server is the sole way the centre in the basic-DML incorporates

the training outcomes from distant workers.

The centre has some extra processing power in the computing server for sub-dataset learning in the semi-DML scenario. As a result, it will retain some sub-datasets and use them for independent learning. In other words, the centre will combine the output from dispersed workers and learn from a subset of datasets in the semi-DML.

DATASET:

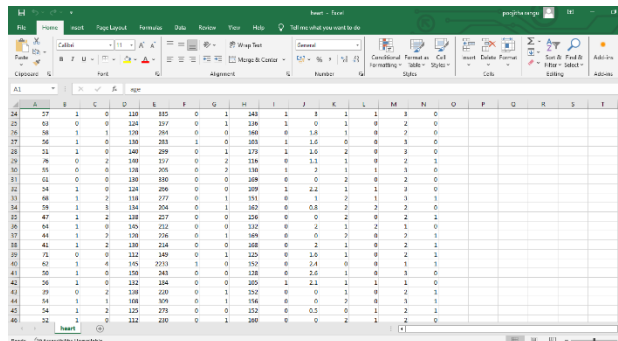
To implement this project, we have used heart disease dataset and in below dataset screen we can see dataset contains invalid data which called as Data Poison.



	age	gender	ttp	trestbps	chol	fbs	resting	trestach	maxng	oldpeak	slope	ca	thal	target
1	39	1	2	130	275	0	1	275	0	0	2	4	2	1
2	50	1	0	144	200	0	0	136	1	0.9	1	0	1	0
3	45	0	2	160	360	0	0	331	0	0.8	2	0	2	1
4	52	1	1	120	205	1	1	284	0	0	2	0	2	1
5	62	1	1	128	208	1	0	340	0	0	2	0	2	1
6	58	0	0	130	297	0	1	331	0	0.6	1	0	2	1
7	54	0	1	132	288	1	0	339	1	0	2	1	2	1
8	48	0	2	130	275	0	1	339	0	0.2	2	0	2	1
9	44	1	0	122	290	0	0	353	0	0	2	1	2	0
10	77	1	0	125	304	0	0	362	1	0	2	3	2	0
11	60	1	0	130	253	0	1	344	1	1.4	2	1	1	0
12	71	0	1	160	302	0	1	362	0	0.4	2	2	2	1
13	70	1	0	145	174	0	1	325	1	2.6	0	0	3	0
14	59	1	2	150	232	1	1	337	0	1.6	2	0	2	1
15	45	0	0	138	238	0	0	352	1	0.2	1	0	2	1
16	64	1	3	170	227	0	0	355	0	0.6	1	0	3	1
17	79	1	1	130	264	0	0	282	0	0	2	0	2	1
18	54	0	2	150	267	0	0	367	0	0	2	0	2	1
19	58	1	0	138	234	0	0	331	1	2.2	1	3	3	0
20	64	1	0	128	263	0	1	305	1	0.2	1	1	3	1
21	63	0	0	108	269	0	1	369	1	1.6	1	2	2	0
22	46	0	2	120	311	0	0	335	0	1.5	1	0	2	1

In above screen heart dataset first row contains column names and remaining

rows are the column values and in below dataset screen we can see odd or invalid value



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
24	27	1	0	230	185	0	1	140	1	1	1	1	1	1	0					
25	63	0	0	124	217	0	1	238	1	0	1	0	0	0	0					
26	58	1	1	130	261	0	0	140	0	1.8	1	0	0	0	0					
27	58	1	0	130	261	1	0	140	1	1.0	0	0	0	0	0					
28	51	1	0	140	209	0	1	175	1	1.0	2	0	0	0	0					
29	26	0	2	140	117	0	2	116	0	1.1	1	0	2	1	0					
30	55	0	0	136	205	0	2	130	1	2	1	1	1	0	0					
31	61	0	0	130	180	0	0	148	0	0	2	0	0	0	0					
32	34	1	0	124	206	0	0	109	1	2.2	1	1	1	0	0					
33	48	1	2	138	177	0	1	151	0	1	2	1	1	1	0					
34	39	1	1	134	204	0	1	142	0	0.8	2	2	2	0	0					
35	47	1	2	138	217	0	0	148	0	0	2	0	0	0	0					
36	84	1	0	140	212	0	0	112	0	2	1	2	1	0	0					
37	41	1	2	136	176	0	1	148	0	0	2	0	0	0	0					
38	41	1	2	130	214	0	0	148	0	2	1	0	2	1	0					
39	71	0	0	112	140	0	1	125	0	1.0	1	0	2	1	0					
40	62	1	0	105	213	1	0	152	0	1.4	0	0	1	1	0					
41	50	1	0	150	141	0	0	116	0	2.0	1	0	1	0	0					
42	56	1	0	132	184	0	0	207	1	1.2	1	1	1	0	0					
43	39	0	2	138	220	0	1	152	0	0	1	0	2	1	0					
44	14	1	1	106	109	0	1	104	0	0	2	0	1	1	0					
45	54	1	2	137	173	0	0	152	0	0.5	0	1	2	0	0					
46	32	1	0	132	220	0	1	100	0	0	2	1	2	0	0					

In above screen in selected blue value, we can see recorded blood pressure value as 2233 which is wrong value and if ML train on such data then it may predict wrong result and it will reduce prediction accuracy and to avoid such problem, we can apply Data Poison Detection technique. In python we can 'Isolation Forest' class to detect and remove such poison data.

DISTRIBUTED MACHINE LEARNING

A method called distributed machine learning (DML) is used to train machine learning models on big datasets that are too big for a single system to handle in a reasonable amount of time. In DML, the dataset is

divided into many smaller subsets and dispersed over several computers so that the model may be trained in parallel. The use of DML has several advantages. Initially, it makes it feasible to analyses enormous datasets that would be impractical to manage on a single system. Second, by enabling many machines to operate simultaneously, it may shorten the model's training period. Thirdly, by offering a more varied and representative collection of training data, it helps raise the model's accuracy and resilience. Model parallelism and data parallelism are the two primary forms of DML. Data parallelism uses separate subsets of the data for each machine's training, with the combined results training the final model. Model parallelism divides the model into smaller parts, then trains one or more of those parts on each machine.

CHALLENGES IN DISTRIBUTED MACHINE LEARNING (DML)

Keeping the machines' communication under control and making sure the training process is well-organized are two major issues in distributed machine learning. This necessitates giving careful thought to elements like load balancing techniques, communication protocols, and network layout.

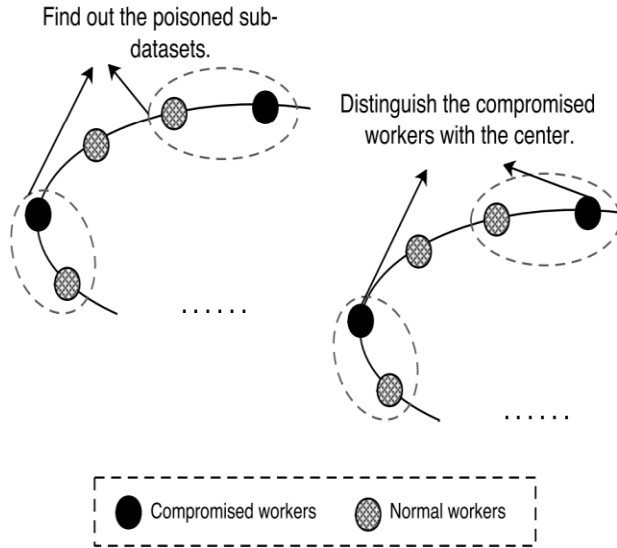
In distributed machine learning, handling inconsistent and heterogeneous input presents another difficulty. The data may have various properties, biases, and distributions since it is spread across many devices. This may result in biased final models and inconsistent learning processes. In order to overcome this, methods like as feature selection, data balance, and data normalization may be used to guarantee that the data is consistent and representative across all computers. Distributed machine learning presents additional security concerns and difficulties on top of these already existing difficulties. For

instance, bad actors can try to steal confidential information, undermine network security, or introduce tainted data into the training process. Implementing efficient security and privacy protections like data encryption, access restriction, and data poisoning detection techniques is necessary to mitigate these dangers. Distributed machine learning has several uses in a wide range of industries, including computer vision, natural language processing, healthcare, and many more, despite these difficulties.

DATA POISON DETECTION SCHEME IN SEMI-DML

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. and the algorithm is described in Algorithm. 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 cannot be realized in the DML scenario.



i). OPTIMAL ALLOCATION SCHEME FOR THE SEMI-DML SCENARIO

The wasted resource W in the system is related to three parameters: p , β and R , where p is the compromised probability of a distributed worker β 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:

$$W(p, \beta, R)$$

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

where S is the number of sub-datasets learned on the distributed workers, and $n(p, S)$ is the number of compromised workers when there are N_w workers in total.

Algorithm Data Poison Detection Scheme of Semi-DML

Input:

Set of workers $E = \{f_i | i \in \{1, \dots, T\}\}$
Set of sub-datasets $\{e_m | m \in \{1, \dots, T\}\}$
Training results of sub-dataset $e_n: p_n = \{w_{n,i}, w_{n,j}\}$
Set of suspicious workers: $W_{sus} = \emptyset$

1: for $n = 1 : T$ do

% Relearn the poisoned sub-dataset by the center.

2: if $\|w_{n,i} - w_{n,j}\|_2 \geq \epsilon$ then

3: Train e_m in the center and get w_n ;

% Find the compromised workers.

4: if $\|w_n - w_{n,i}\|_2 \geq \epsilon$ then

5: Add f_i to W_{sus} ;

6: end if

7: if $\|w_n - w_{n,j}\|_2 \geq \epsilon$ then

8: Add f_j to W_{sus} ;

9: end if

10: end if

11: end for

Output:

Set of suspicious workers: W_{sus} ;

VI. MODULE DESCRIPTION:

The modules below are what we created in order to carry out this project.

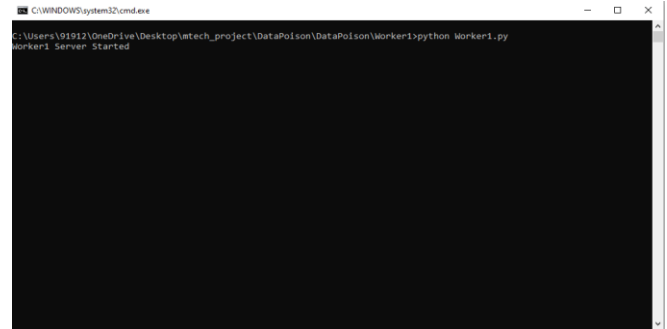
Worker1: This worker node receives split datasets from the centre server, builds the SVM and Basic DML models that are already in place, calculates the accuracy of both methods, and sends the results back to the centre server.

Worker2: This additional worker node receives the second half of the dataset, runs the SVM and Basic DML that are already in place, and then sends the accuracy back to the central server.

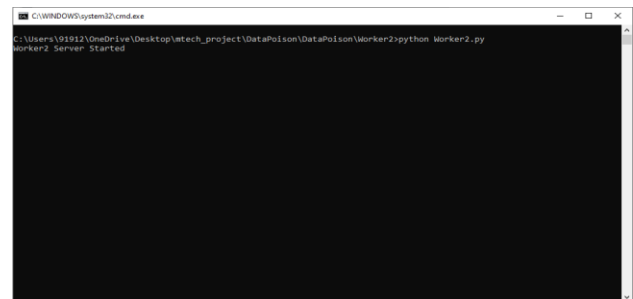
Centre Server: The centre server uploads the dataset to the application, splits it into two equal halves, distributes each portion to workers 1 and 2, and then gathers the results. This server will compute accuracy and execute semi-DML.

VII. RESULT:

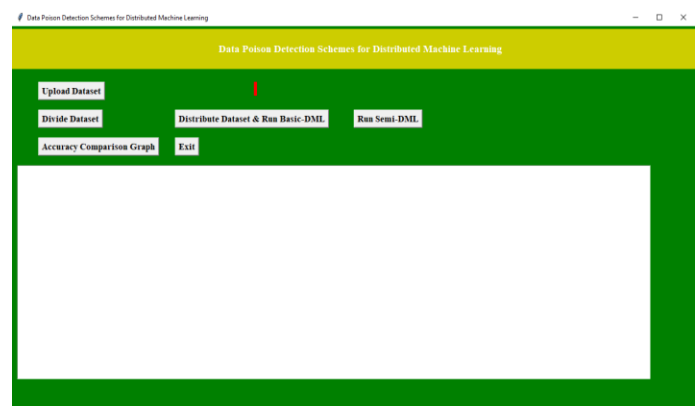
To run project first double click on 'run.bat' file from Worker1 folder to start worker 1 node and to get below screen



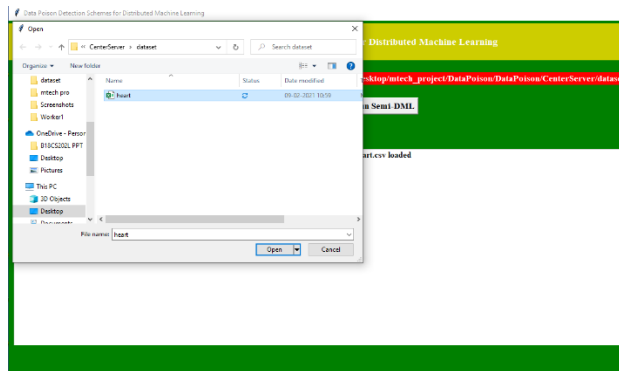
In above screen worker 1 server started and now double click on 'run.bat' file from worker2 folder to start worker 2



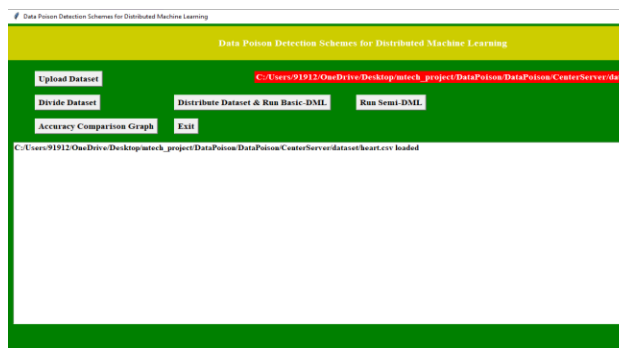
In above screen worker2 server started and now double click on 'run.bat' file from 'CenterServer' folder to start distributed server and to get below screen



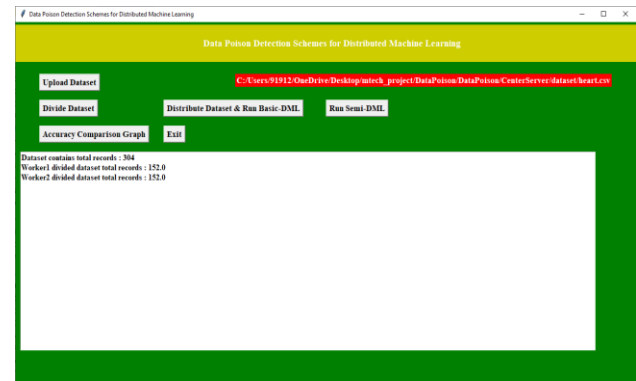
In above screen click on 'Upload Dataset' button to upload dataset and to get below screen



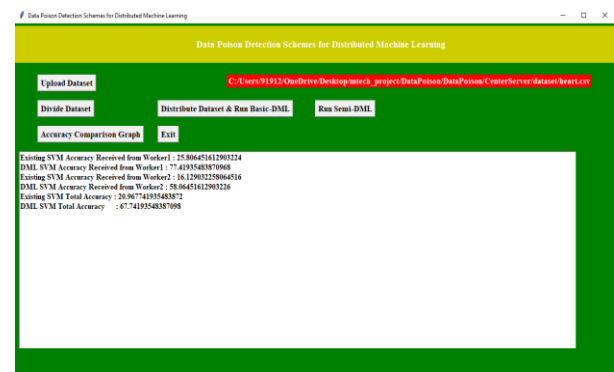
In above screen selecting and uploading 'heart.csv' file and then click on 'Open' button to load dataset and to get below screen



In above screen dataset loaded and now click on 'Divide Dataset' button to divide dataset into 2 equal parts

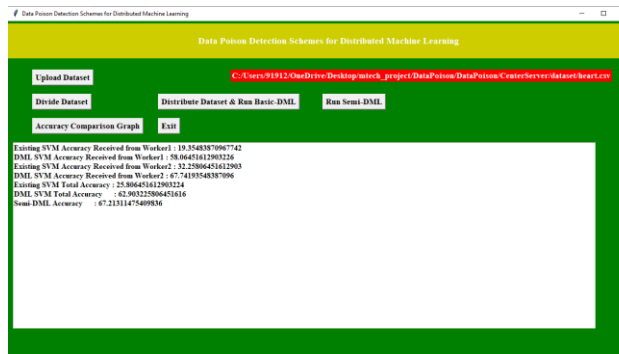


In above screen dataset contains 304 records and equally distributed to 2 parts and now click on 'Distribute Dataset & Run Basic-DML' button to distribute dataset to 2 workers and then get accuracy result

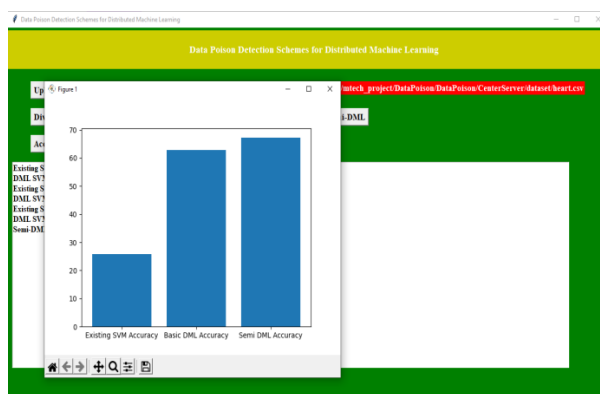


In above screen we got result from 2 worker nodes for existing SVM accuracy and propose DML accuracy and in above screen we can see existing SVM accuracy is 20% when data poison exists in dataset and after removing data poison using DML

technique we got 67% accuracy and now click on 'Run Semi-DML' button to allow center server to devote resources to DML and then remove poison from dataset and then calculate accuracy



In above screen Semi-DML accuracy is 67% 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



Data Poisoning Scheme	Accuracy (%)
Basic DML	62.90322806451616
Semi-DML	67.21311475409936
Existing SVM	25.806451612903214

VIII. CONCLUSION

In this research, we spoke about basic-DML and semi-DML data poisoning detection algorithms. In the basic-DML scenario, the data poison detection strategy uses a threshold of parameters to identify the sub-datasets that are poisoned. Additionally, we developed a mathematical model to examine the likelihood of detecting threats at various training loop counts. Additionally, we demonstrated the best resource allocation in the semi-DML situation as well as an enhanced data poisoning detection technique. According to simulation data, the

suggested technique may improve model correctness by 20%–30% in the basic-DML scenario. When compared to the other two methods that do not have optimum resource allocation, the enhanced data poison detection technique with optimal resource allocation may reduce resource waste by 50–60% in the semi-DML scenario.

IX. FUTURE ENHANCEMENT

The data poisoning detection strategy may be expanded to a more dynamic pattern in the future to accommodate the evolving application environment and level of assault. Furthermore, further research is required on the trade-off between resource cost and security since multi-training sub-datasets will raise the system's resource consumption.

X. 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. Muñoz-González, 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.