INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

DATA POISON DETECTION SCHEMES FOR DISTRIBUTED MACHINE LEARNING

[1]Dr.Ch. V. PHANI KRISHNA, [2]BALASANI DINESH, [3]VEMULA VAMSHI, [4]YELDANDI VENKAT

[1]Assistant Professor, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad,

[2,3,4]BTech Student, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad

balasanidinesh7@gmail.com, vamshivemula444@gmail.com,
venkatyeldandi2000@gmail.com

Abstract: *Distributed Machine Learning (DML), which is used when a single node cannot accurately process massive datasets within an acceptable time. However, this will inevitably expose more potential targets to attackers compared with the non-distributed environment. In this project, 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.*

Keywords: *Distributed Machine Learning, distributed environment, Data poison detection.*

I. INTRODUCTION

Distributed machine learning (DML) has been widely used in distributed systems where no single node can get the intelligent decision from a massive dataset within an acceptable time. In a typical DML system, a central server has a tremendous amount of data at its disposal. It divides the dataset into

ISSN: 2057-5688

different parts and disseminates them to distributed workers who perform the training tasks and return their results to the center. 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[1].

Poisoning attack 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. initially demonstrated that attackers could manipulate the data to defeat the data miner if they have complete information. Then Lowd et al. 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, focusing on non-distributed machine learning context. Recently, there are a couple of efforts devoted in preventing data from being manipulated in DML[2].

For example, Zhang et al. and Esposito et al. 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 project, we classify DML into basic distributed machine learning (basic-DML) and semi distributed machine learning (semi-DML), depending on whether

ISSN: 2057-5688

the center shares resources in the dataset training tasks. Then, we present data poison detection schemes for basic-DML and semi-DML respectively.

With the rapid advances in technology, the use of computers and the data generated by these systems has increased significantly. As the rate of data generated increases, the computer gains direct access to the data and can use this generated data without further programming. The computer can give meaningful results by learning from its data. They are provided by machine learning techniques, which are artificial intelligence applications currently used in cyber security. In addition, with the rise of cybercrime, machine learning techniques are being used to detect malicious behaviour of computers, malware, and malicious traffic on the network. Two opposite mechanistic learning methods depend on the time of the attack. Pre-sample training data poisoning: An attacker changes the label of the training dataset before the sample is trained. Data preparation based on the trained model: After training the model, the attacker forces the model to create an interaction with the actual output data. Both attacks are very dangerous with consequences and impacts. Examining off-the-shelf machine learning products reveals that data addiction attacks pose an even bigger threat. Almost all commercial products require training datasets from the installation company. Attackers can easily poison this database[3].

Figure 1.1 shows the malware can be placed in a company where the malware determines the attack time and what the best attack vector is. These attacks vary by design, making the detection very difficult and lengthy

ISSN: 2057-5688

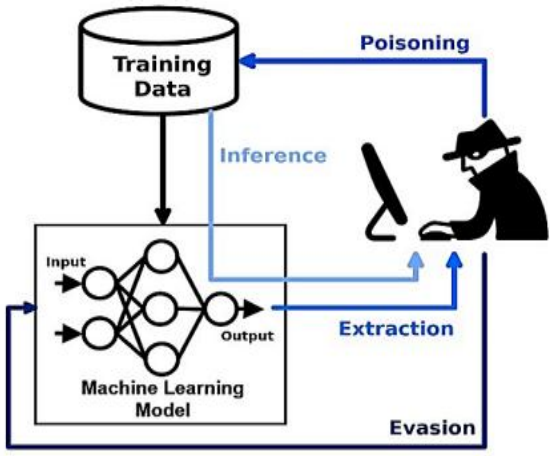

Fig.1 Data poisoning attack

II. LITERATURE SURVEY

G. Qiao et al. [4] Mobile edge computing (MEC) has 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.

K. Zhang et al.[5] The Internet of Things (IoT) platform has played a significant role in improving road

ISSN: 2057-5688

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 characteristics of communication modes and spectrum resources chosen by vehicles in different 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.

T. Chen et al. [6] MXNet is a multi-language 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

ISSN: 2057-5688

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.

L. Zhou et al. [7] 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 advert 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 big data, user, domain, and system. The phases 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 or under explored research areas.

J. Chen et al. [8] described Deep Poison as 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 Poison is that the toxic training models developed cannot be distinguished from harmless ones by defensive methods or human visual inspection, and even harmless test models can be attacked.

C. Li et al. [9] described, Machine Learning (ML) is widely used to detect malware on various platforms,

ISSN: 2057-5688

including Android. Detection models must be retested following the data collected (e.g., monthly) to continue 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 the Android Malware Detector.

J. Chen et al. [10] 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 defences 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 been developed.

III. PROPOSED SYSTEM

In distributed environment sometime attackers may modify training data and then make ML to predict wrong result and to detect and remove such modified data we are using Data Poison Detection technique. This technique will inspect training dataset to identify odd values and then remove it. By applying Data Poison technique, we can improve accuracy of ML algorithms.

Basic-DML and Semi-DML

In this project we are using two distributed techniques called Basic DML and Semi DML as shown in below figure. Basic DML will divide dataset into multiple parts and send to worker nodes and worker nodes build ML model and send result back distributed center server. In Semi DML center server itself will devote resource to ML model to train dataset.

SYSTEM ARCHITECTURE

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

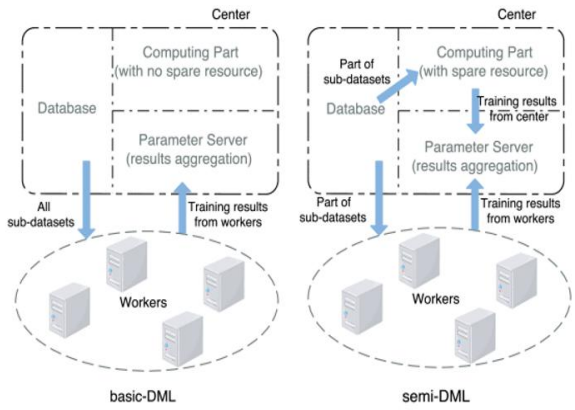

Fig.1 System architecture

What is Machine Learning:

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Leaning: -

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modelling the relationship between measured features of data and some label associated with the

ISSN: 2057-5688

data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modelling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Challenges in Machines Learning:

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are:

•**Quality of data**: Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

•**Time-Consuming task:** Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

•**Lack of specialist persons**: As ML technology is still in its infancy stage, availability of expert resources is a tough job.

•No clear objective for formulating business problems: Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

ISSN: 2057-5688

•Issue of overfitting & underfitting: If the model is overfitting or underfitting, it cannot be represented well for the problem.

•Curse of dimensionality: Another challenge ML model faces is too many features of data points. This can be a real hindrance.

•Difficulty in deployment: Complexity of the ML model makes it quite difficult to be deployed in real life.

Distributed Machine Learning

Distributed machine learning (DML) is a technique used to train machine learning models on large datasets that cannot be processed by a single machine within a reasonable time frame. In DML, the dataset is split into multiple smaller subsets and distributed across multiple machines, which work in parallel to train the model.

There are several benefits to using DML. First, it allows for the processing of large datasets that would otherwise be impossible to handle on a single machine. Second, it can reduce the training time of the model by allowing multiple machines to work in parallel. Third, it can improve the accuracy and robustness of the model by providing a more diverse and representative set of data for training.

There are two main types of DML: data parallelism and model parallelism. In data parallelism, each machine trains on a different subset of the data, and the results are combined to train the final model. In model parallelism, the model is split into smaller components, and each machine trains one or more components of the model.

Challenges in Distributed Machine Learning (DML)

One of the key challenges in distributed machine learning is managing the communication between the machines and ensuring that the training process is coordinated and efficient. This requires careful consideration of factors such as the network topology, communication protocols, and load balancing strategies.

ISSN: 2057-5688

Another challenge in distributed machine learning is dealing with data heterogeneity and inconsistency. Since the data is distributed across multiple machines, it may have different characteristics, biases, and distributions. This can lead to inconsistencies in the learning process and biases in the resulting model. To address this, techniques like data normalization, data balancing, and feature selection can be used to ensure that the data is representative and consistent across all machines.

In addition to these challenges, distributed machine learning also introduces new security risks and challenges. For example, malicious actors may attempt to inject poisoned data into the training process, compromise the security of the network, or steal sensitive data. To address these risks, effective security and privacy measures such as data encryption, access control, and data poison detection schemes must be implemented.

Despite these challenges, distributed machine learning has many applications in various fields such as natural language processing, computer vision, healthcare and many more fields.

IV. RESULTS

Worker 1 Interface:

To start the worker 1, first double click on 'run.bat' file from Worker1 folder.

Fig.2 Worker 1 Interface

Worker 2 Interfaces:

ISSN: 2057-5688

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

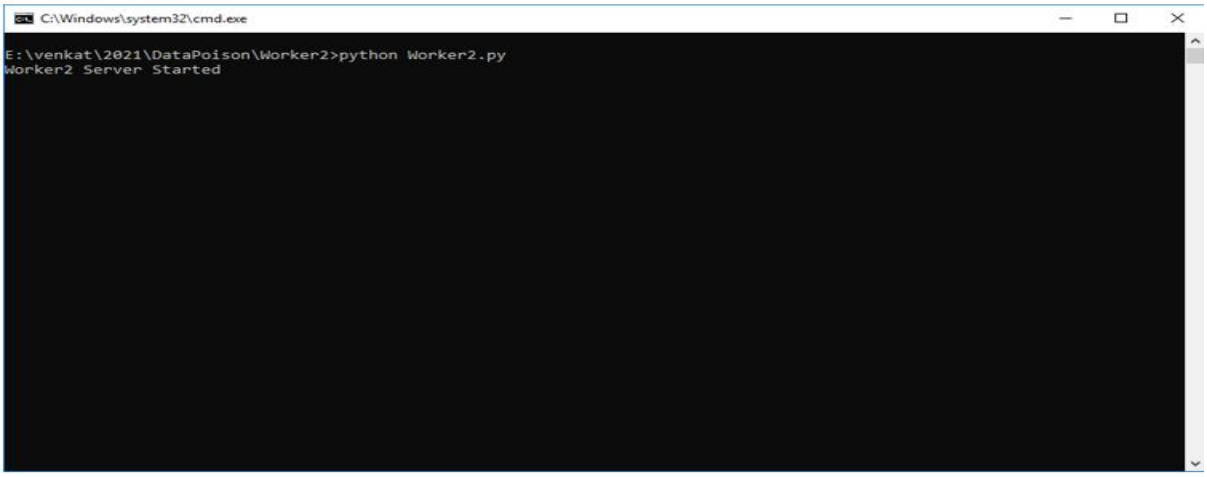

Fig.3 Worker 2 Interface

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. Center Server has following options:

- Upload Dataset
- Divide Dataset
- Distribute and run Basic DML
- Run Semi DML
- Accuracy Comparison Graph

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

Fig.4 Center Server Interface

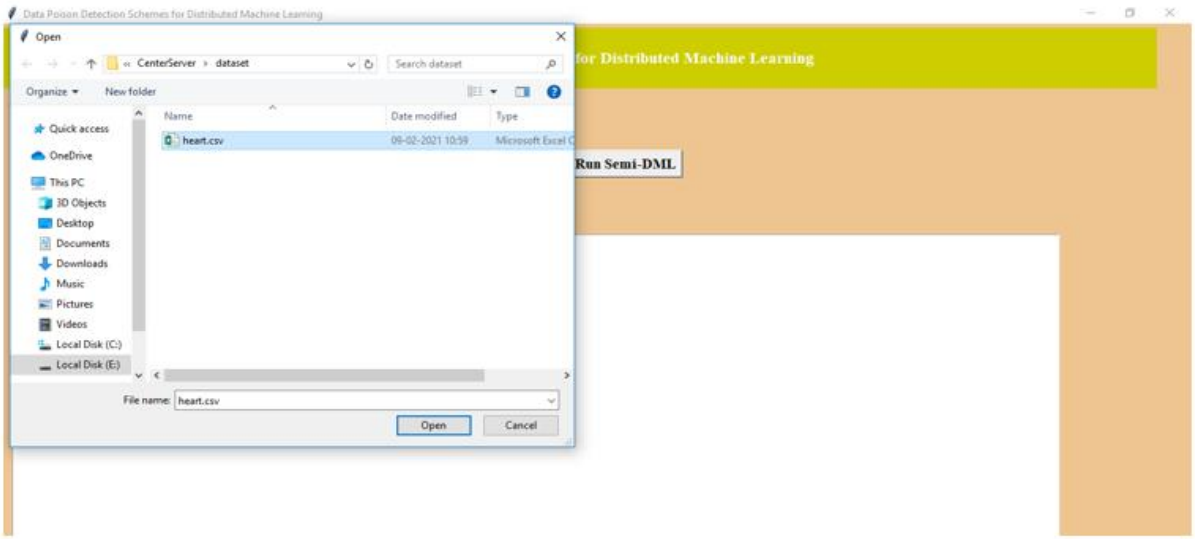

Fig.5 Uploading Dataset

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

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

Fig.6 Uploading dataset

In above screen dataset loaded and now click on 'Divide Dataset' button to divide dataset to two worker nodes

Fig.7 Dividing dataset

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.

ISSN: 2057-5688

Fig.8 Run Basic DML

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 19% when data poison exists in dataset and after removing data poison using DML technique we got 51% accuracy.

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

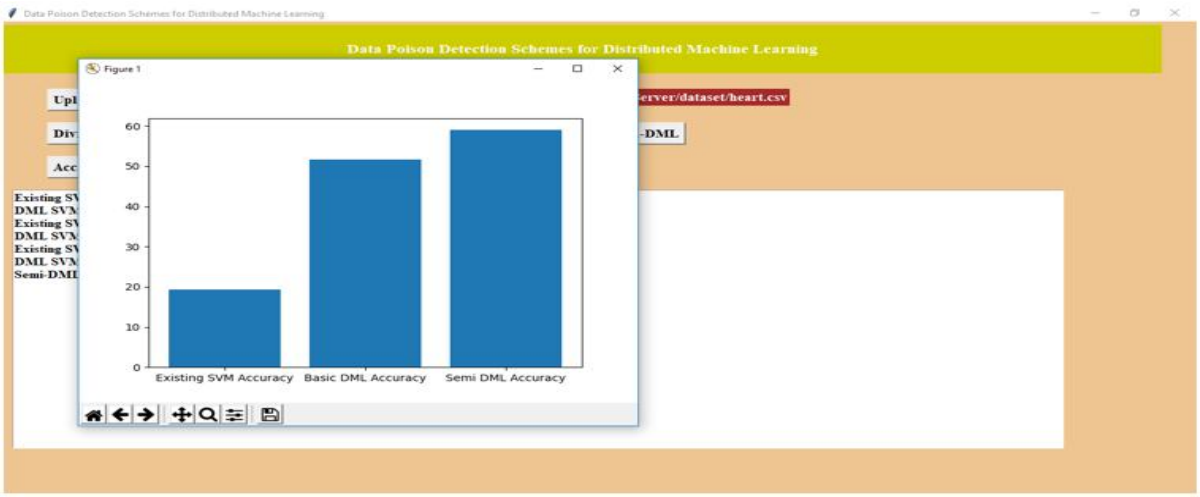

Fig.9 Accuracy comparison

V. CONCLUSION

In this project, 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% - 40%. As to the semi-DML scenario, the

ISSN: 2057-5688

improved data poison detection scheme with optimal resource allocation can decrease the resource wastage by 30-50% compared to the other two schemes without the optimal resource allocation.

REFERENCES

[1] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," IEEE Communications Magazine, vol. 56, no. 8, pp. 48–54, 2018.

[2] K. Zhang, S. Leng, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1987–1997, 2019.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning." in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), vol. 16. USENIX Association, 2016, pp. 265–283.

[4] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," CoRR, vol. abs/1512.01274, 2015.

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

[6] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for big data: A survey," IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 531–549, 2017.

[7] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server." in 11th

ISSN: 2057-5688

USENIX Symposium on Operating Systems Design and Implementation (OSDI), vol. 14. USENIX Association, 2014, pp. 583–598.

[8] B. Fan, S. Leng, and K. Yang, "A dynamic bandwidth allocation algorithm in mobile networks with big data of users and networks," IEEE Network, vol. 30, no. 1, pp. 6–10, 2016.

[9] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home m2m networks: Architectures, standards, and qos improvement," IEEE Communications Magazine, vol. 49, no. 4, pp. 44–52, 2011.

[10] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, and Y. Zhang, "Blockchain and deep reinforcement learning empowered intelligent 5g beyond," IEEE Network Magazine, vol. 33, no. 3, pp. 10–17, 2019.

[11] Prasadu Peddi (2017) "Design of Simulators for Job Group Resource Allocation Scheduling In Grid and Cloud Computing Environments", ISSN: 2319- 8753 volume 6 issue 8 pp: 17805-17811.