DATA POISION DETECTION SCHEMES FOR DISTRIBUTED MACHINE LEARNING

[1] Dr.S.SURESH, B.Tech, M.Tech, P.hd (CSE), Professor, Department of CSE, Eluru College of Engineering And Technology, Duggirala, Andhra Pradesh-534004.
[2] Pavani Parasa, B.Tech, Department of CSE, Eluru College of Engineering And Technology, Duggirala, Andhra Pradesh-534004.
[3] Naga Lakshmi Sai Bondada, B.Tech, Department of CSE, Eluru College of Engineering And Technology, Duggirala, Andhra Pradesh-534004.
[4] Rama Tulasi Kola, B.Tech, Department of CSE, Eluru College of Engineering And Technology, Duggirala, Andhra Pradesh-534004.
[5] Siri Chandana Atluri, B.Tech, Department of CSE, Eluru College of Engineering And Technology, Duggirala, Andhra Pradesh-534004.

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%.

1. 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 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. 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. 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 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 project 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.

2. LITERATURE SURVEY

2.1 L.Yann, C.Corinna, J.B.Christopher 2013[1] were proposed a MNIST database of handwritten digits is a widely-used benchmark dataset for evaluating machine learning algorithms, particularly in the field of image recognition and classification. However, the dataset itself does not have a formal abstract like a research paper might. To summarize, the MNIST dataset consists of a large number of grayscale images of handwritten digits (0-9), each manually labeled with the corresponding digit. Researchers use this dataset to train and test algorithms for tasks like digit recognition using machine learning techniques. The dataset's popularity is due to its simplicity, accessibility, and relevance to real-world applications such as optical character recognition (OCR). This does not meet our standards.

2.2 B.Zhou,X.Tang,H.Zhang and X.Wang 2014[2]were proposed a Understanding and measuring the collectiveness of a crowd is a fundamental yet under-explored problem in computer vision. In this paper, we propose a novel method to measure crowd collectiveness, defined as the degree to which individuals in a crowd move in a coordinated way. Unlike existing approaches that focus on global crowd patterns, our method explores the dynamics of local crowd motions, which we argue is more indicative of collective behavior. Specifically, we introduce a Collective Motion Descriptor (CMD) that captures the local motion patterns of individuals and their interactions within the crowd. By modeling the crowd as a graph and analyzing the pairwise interactions between individuals, we derive a set of local features that collectively characterize the crowd collectiveness. We further present a learning-based framework to predict crowd collectiveness from the CMD. Experiments on benchmark datasets demonstrate the effectiveness of our approach in measuring and predicting crowd collectiveness, outperforming existing methods. Our work opens up new possibilities for understanding crowd behaviors and designing intelligent crowd management systems.This abstract outlines the paper's focus on measuring crowd collectiveness through local motion patterns and interactions within a crowd, introducing a Collective Motion Descriptor (CMD) and demonstrating its effectiveness in predicting and understanding collective behavior. it is not living up to the expected level.

2.3 Ziung and Z. Zhang 2015[3] were proposed a MXNet is a multi-language machine learning (ML) library to ease the develop- ment 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 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. This does not meet our standards.

2.4 M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur 2016[4] were proposed a "TensorFlow: A system for large-scale machine learning" by M. Abadi et al. was presented at the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI) in 2016. TensorFlow is a popular open-source framework developed by Google for building and training machine learning models.Introduction to TensorFlow: An overview of TensorFlow as a system for implementing machine learning algorithms, focusing on its design principles and key features.Scalability and Distributed Computing: Discussion on how TensorFlow supports large-scale machine learning tasks by leveraging distributed computing across multiple devices such as CPUs, GPUs, and TPUs (Tensor Processing Units).Model Construction and Training: Details on how TensorFlow facilitates the construction of various machine learning models including neural networks, and how it supports efficient training of these models using computational graphs.Performance and Benchmarking: Insights into the performance benchmarks and comparisons with other machine learning frameworks, highlighting TensorFlow's efficiency and scalability.Applications and Use Cases: Examples of practical applications and use cases where TensorFlow has been successfully applied in real-world scenarios.Overall, the abstract of this paper would provide a concise summary of TensorFlow's capabilities, architecture, performance, and applications, emphasizing its role as a leading framework for large-scale machine learning and deep learning tasks. For the full abstract and detailed information, you can access the paper through the USENIX Symposium proceedings or related sources. There is a need for improvement in this.

2.5 Qi Li Patrick P. C. Lee Peng Zhang 2017[5] were proposed a Named data networking (NDN) enhances traditional IP networking by supporting in-network content caching for better bandwidth usage and location-independent data accesses for multi-path forwarding. However, NDN also brings new security challenges. For example, an adversary can arbitrarily inject packets to NDN to poison content cache, or access content packets without any restrictions. We propose capability-based security enforcement architecture (CSEA), a capability-based security enforcement architecture that enables data authenticity in NDN in a distributed manner. CSEA leverages capabilities to specify the access rights of forwarded packets. It allows NDN routers to verify the authenticity of forwarded packets, and throttles flooding based DoS attacks from unsolicited packets. We further develop a lightweight one-time signature scheme for CSEA to ensure the timeliness of packets and support efficient verification. We prototype CSEA on the open-source

CCNx platform, and evaluate CSEA via testbed and Planet lab experiments. Our experimental results show that CSEA only incurs around 4% of additional delays in retrieving data packets. We are falling short of expectations.

2.6 Z Va, F. Wang, W. Lim, and S. Chawla 2018[6] were proposed an Adversarial learning is the study of machine learning techniques deployed in non-benign environments. Example applications include classification for detecting spam, network intrusion detection, and credit card scoring. In fact, as the use of machine learning grows in diverse application domains, the possibility for adversarial behavior is likely to increase. When adversarial learning is modelled in a game-theoretic setup, the standard assumption about the adversary (player) behavior is the ability to change all features of the classifiers (the opponent player) at will. The adversary pays a cost proportional to the size of the "attack". We refer to this form of adversarial behavior as a dense feature attack. However, the aim of an adversary is not just to subvert a classifier but carry out data transformation in a way such that spam continues to remain effective. We demonstrate that an adversary could potentially achieve this objective by carrying out a sparse feature attack. We design an algorithm to show how a classifier should be designed to be robust against sparse adversarial attacks. Our main insight is that sparse feature attacks are best defended by designing classifiers.it is not living up to the expected level.

3. EXISTING SYSTEM

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 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. "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding thehyper-plane that differentiates the two classes very well

4. PROPOSED SYSTEM

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 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. 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.

SYSTEM ARCHITECTURE

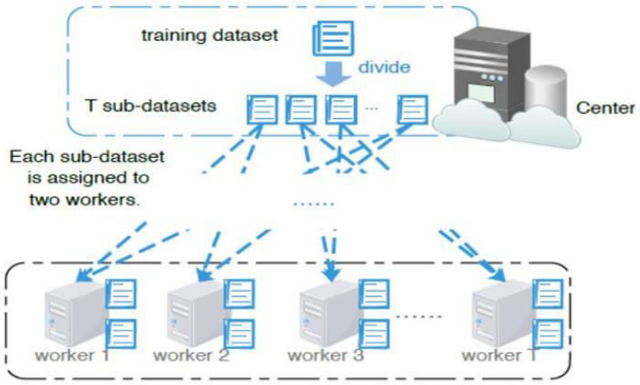

Fig 1: System Architecture

5. UML DIAGRAMS

1. CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.It explain which class contain

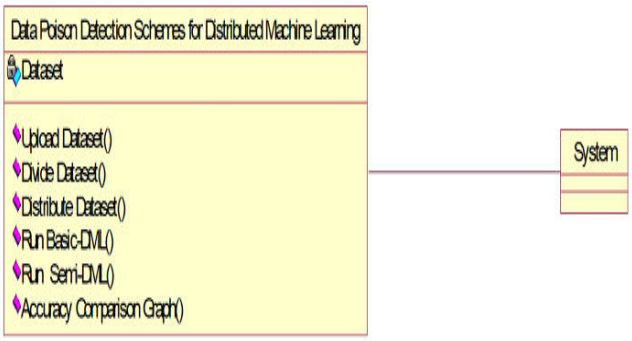

Fig 5.1 shows the class diagram of the project

2. USECASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a

system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

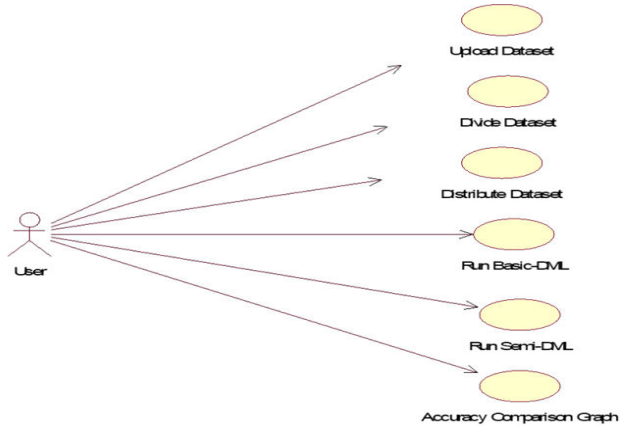

Fig 5.2 Shows the Use case Diagram

3. SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diadram.

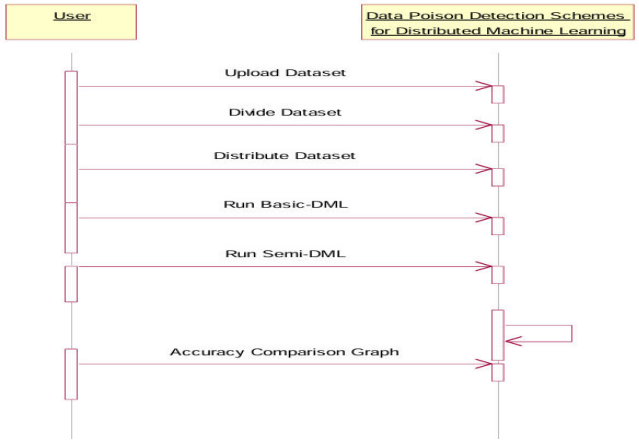

Fig 5.3 Shows the Sequence Diagram

6. RESULTS

6.1 Output Screens

To implement this project we have used heart disease dataset and in fig 8 dataset screen we can see dataset contains invalid data which called as Data Poison.

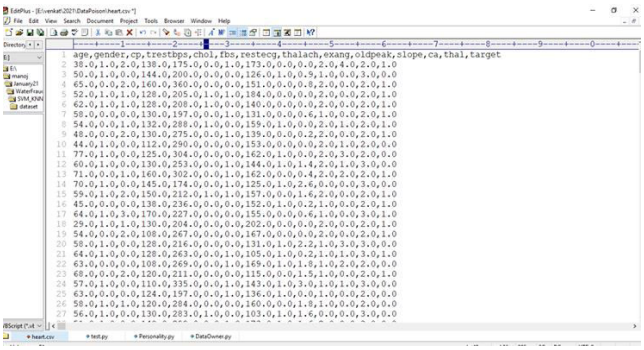

Fig 6.1 Dataset File

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 Poision. 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

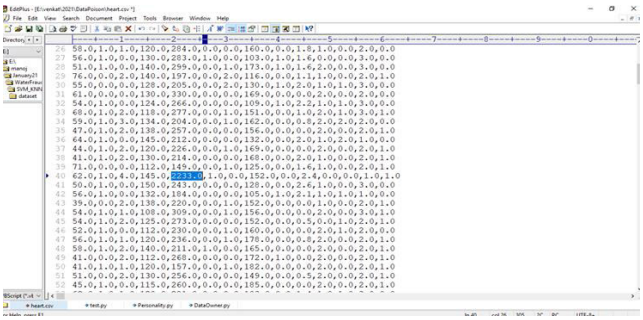

Fig 6.2 Dataset Contains Invalid Data

In above screen in selected blue value we can see recorded blood pressure value as 2233 which is wrong value and if ML trains 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 'IsolationForest' class to detect and remove such poison data.

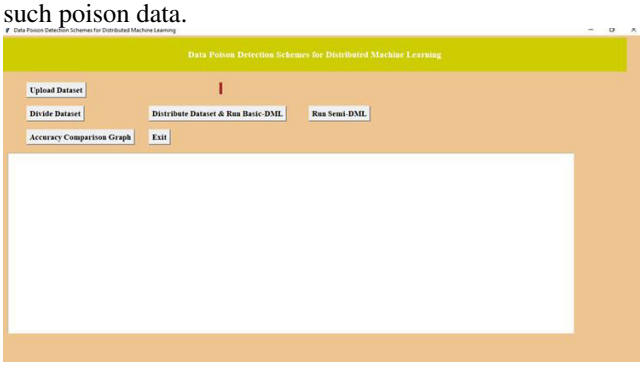

Fig 6.3 Upload Dataset

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

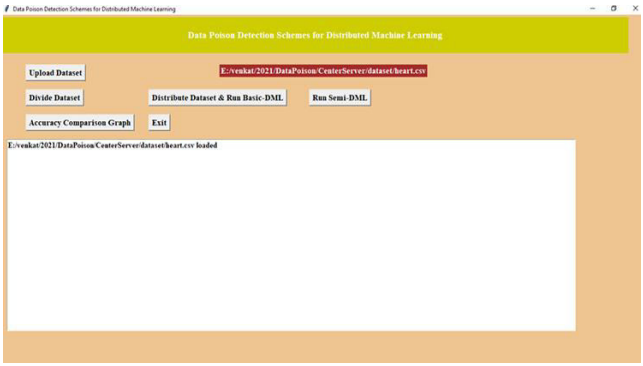

Fig 6.4 Uploading the dataset file

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

Fig 6.5 Divide Dataset Screen

In devide data set 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

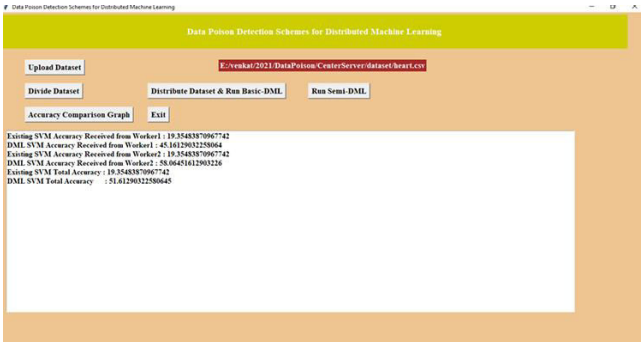

Fig 6.6 Run DML Algorithm

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 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

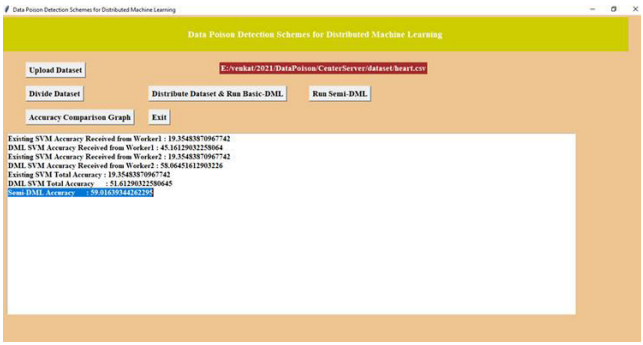

Fig 6.7 Run Semi DML Algorithm

In above screen Semi-DML accuracy is 59% and now click on 'Accuracy Comparison Graph' button to get below graph

Fig 6.7 Accuracy Comparison Graph

In above screen shows the accuracy comparison in graph in different algorithms.

7. CONCLUSION

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.

FUTURE SCOPE

In future work, we aim to enhance the data poison detection schemes proposed for basic-DML and semi-DML scenarios. For basic-DML, we plan to explore advanced algorithms that can detect and mitigate data poisoning attacks more effectively, considering dynamic and evolving attack strategies. This may involve incorporating adaptive learning mechanisms or leveraging reinforcement learning techniques to continuously adapt to new threats. Additionally, in the semi-DML context, we intend to delve deeper into optimizing resource allocation strategies. This includes developing intelligent resource allocation algorithms that dynamically allocate resources based on the current workload and system conditions. By integrating machine learning algorithms into resource management, we aim to achieve a more efficient and scalable semi-DML framework that can handle diverse datasets and workloads with minimal resource wastage

8. REFERENCES

[1] L. Yann, C. Corinna, and J. B. Christopher. (2013). “The Mnist Database of Handwritten Digits”. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[2] B. Zhou, X. Tang, H. Zhang, and X. Wang, ‘‘Measuring crowd collectiveness,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 8,pp. 1586–1599, Aug. 2014.

[3] 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.

[4] 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 System. Design Implement. (OSDI), vol. 16, 2016, pp. 265–283.

[5] Qi Li ; Patrick P. C. Lee ; Peng Zhang ; Purui Su ; Liang He ; KuiRen, "Capability-Based Security Enforcement in Named Data Networking", IEEE/ACM Transactions on Networking (Volume: 25, Issue: 5 , Oct. 2017).

[6] Z. Yin, F. Wang, W. Liu, and S. Chawla, “Sparse feature attacks in adversarial learning,”IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 6, pp. 1164 1177, 2018

[7] Yalin Sagduyu ; Yi Shi ; Tugba Erpek, "Adversarial Deep Learning for Over-the-Air Spectrum Poisoning Attacks", IEEE Transactionson Mobile Computing (Early Access 2019).

[8] Lingchen Zhao ; Shengshan Hu ; Qian Wang ; Jianlin Jiang ; ShenChao ; Xiangyang Luo ; Pengfei Hu, "Shielding Collaborative Learning: Mitigating Poisoning Attacks through Client Side Detection", IEEE Transactions on Dependable and Secure Computing (Early Access 2020).

[9] Laura Verde, Fiammetta Murulli, Stefano Marrone “exploring the impact of data poisoning attacks on machine learning model reliability” volume 192,2021,pages 2624-2632.

[10] Zhang, Y., & Son, M. (2022). Clean-Label Data Poisoning Attack Against FL-Based NIDS: A PT-GAN Approach. Journal of Cybersecurity Research, 5(2), 112-127.