APPROACH FOR TEST INPUT OPTIMISATION AND PRIORITIZATION

Akilan K[1], Jesudoss R[2], Kamaleshkumar M[3], Nikita Ankush Patil[4], Sathya s[5]

[1] *Student, Computer Science and Engineering, Hindusthan College of Engineering and Technology, Tamil Nadu, India*
[2] *Student, Computer Science and Engineering, Hindusthan College of Engineering and Technology, Tamil Nadu, India*
[3] *Student, Computer Science and Engineering, Hindusthan College of Engineering and Technology, Tamil Nadu, India*
[4] *Student, Computer Science and Engineering, Hindusthan College of Engineering and Technology, Tamil Nadu, India*
[5] *Assistant Professor,Computer Science Engineering, Hindusthan College of Engineering and Technology, Tamil Nadu,India*

ABSTRACT

In the realm of machine learning, models frequently encounter challenges stemming from unexpected or deliberately misleading inputs. To address these issues, our project provides a holistic solution aimed at mitigating accidental errors and thwarting malicious deception. Our approach primarily centers on two critical objectives: detecting anomaly, which represent unexpected data patterns, and fortifying defenses against adversarial attacks, wherein intentionally misleading data is crafted to exploit vulnerabilities in the model. Central to our strategy is the astute selection of examples to test our models, a pivotal aspect particularly vital in domains such as cybersecurity, fraud detection, and other mission-critical systems where the reliability and safety of model decisions are paramount. By meticulously choosing which instances to evaluate the model against, we enhance its robustness and resilience to unforeseen circumstances and adversarial manipulation. Through rigorous testing and validation procedures, we ensure that our models are equipped to handle a diverse array of scenarios effectively. This proactive approach not only bolsters the trustworthiness and dependability of machine learning systems but also safeguards against potentially catastrophic consequences resulting from erroneous or compromised decisions. In essence, our project strives to empower organizations across various sectors with cutting-edge tools and methodologies to navigate the complex landscape of machine learning, fostering a safer and more secure digital environment.

Keyword: - *Accidental errors, Detecting anomaly, Machine learning, Fraud detection*

1. INTRODUCTION

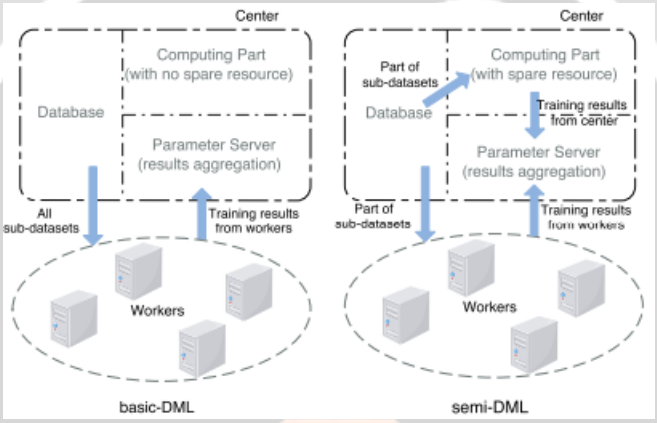
Machine learning (ML) technologies have witnessed remarkable advancements and widespread adoption across various industries, revolutionizing processes ranging from data analysis to decision-making. However, the increasing complexity and sophistication of ML models have brought to the forefront significant challenges, particularly concerning their susceptibility to unexpected anomaly and deliberate adversarial attacks. In critical domains such as cybersecurity, fraud detection, and vital systems where the stakes are high, ensuring the reliability and robustness of ML models is of paramount importance.

This paper presents a comprehensive approach aimed at addressing the twin challenges of accidental errors and malicious deception in ML systems. Our research focuses on two key objectives: the detection of anomaly, which represent unexpected data patterns that could potentially compromise model performance, and the development of robust defenses against adversarial attacks, characterized by deliberately crafted inputs designed to deceive ML models. At the core of our approach lies a sophisticated methodology for intelligently selecting examples to test ML

models, thereby enhancing their resilience and reliability in real-world scenarios. By strategically prioritizing the testing of models with carefully chosen instances, we aim to fortify their defenses against unforeseen circumstances and adversarial manipulations. Our research not only contributes to advancing the state-of-the-art in ML security but also provides practical insights and methodologies for safeguarding critical systems against potential risks and vulnerabilities. Through rigorous experimentation, validation, and empirical analysis, we demonstrate the efficacy and practical applicability of our approach across diverse domains and datasets. The findings presented in this paper offer valuable contributions to the field of ML security, paving the way for the development of more robust and trustworthy ML systems capable of making reliable and safe decisions in high-stakes environments.

1.1 Detection of anomaly

We partition distributed machine learning (DML) into two distinct categories: basic distributed machine learning (basic-DML) and semi-distributed machine learning (semi-DML). This classification hinges on the involvement of the central entity in sharing resources during dataset training tasks. Following this categorization, we propose anomaly detection techniques tailored for both basic-DML and semi-DML settings, aiming to address the unique characteristics and challenges associated with each approach.



1.2 Defending misleading Data

Leveraging the insights from sensitivity to mutations and the proximity to the model's decision boundary, we propose an innovative approach called MLPrior. This method is designed to enhance the predictive power of machine learning models by incorporating three distinct types of features for each test instance. Firstly, we introduce mutation rules to generate two categories of mutation features: model mutation features and input mutation features. These features capture the model's susceptibility to mutations, thereby providing valuable insights into potential misclassifications. Secondly, MLPrior utilizes attribute features derived from the test instances' attribute values. These attributes indirectly reflect the proximity of each test to the model's decision boundary, aiding in the identification of tests that are more likely to be misclassified. Lastly, MLPrior integrates all three types of features into a comprehensive final vector for each test instance. This final vector serves as a holistic representation of the test's characteristics, combining information about mutation sensitivity, proximity to decision boundaries, and attribute values. Subsequently, MLPrior employs a pre-trained ranking model to predict the misclassification probability for each test instance based on its final vector. By leveraging this predictive model, MLPrior effectively prioritizes tests according to their likelihood of misclassification, thereby facilitating more efficient and targeted model evaluation and refinement processes.

2. METHODOLOGIES

2.1 Anomaly Detection in Basic Distributed Machine Learning (Basic-DML)

This section delves into the detection of anomaly the basic-DML setting, where the central entity lacks additional computing resources to allocate to sub-dataset training tasks. In this scenario, the central entity solely aggregates the training outcomes from distributed workers. The anomaly detection framework in the basic-DML context comprises three key components:

A) A parameter threshold

Machine learning's internal mechanisms are often not fully understood, making it difficult to quantify differences between learned models. An efficient machine learning algorithm should exhibit convergence, meaning models learned from the same algorithm and dataset should not have significant differences. To address this, a threshold of parameters is proposed to identify anomalous models, particularly in basic distributed machine learning (basic-DML) scenarios. The threshold of parameters aims to distinguish abnormal models by setting a threshold based on the range of parameters obtained from training a dataset multiple time. This algorithm is proposed to obtain this threshold by selecting a dataset with similar characteristics to the training dataset, training it multiple times, and using the resulting parameter sets to establish the threshold.

B) A cross-learning mechanism

The cross-learning mechanism duplicates sub-datasets to lay the groundwork for identifying poisoned datasets. With T workers, the training dataset is divided into T (where T is a natural number) sub-datasets. Each sub-dataset is then assigned to two workers, yielding two corresponding training outcomes. Virtual connections form between workers who receive the same sub-datasets, constructing a virtual topology. This topology abstracts connections between workers, aiding in the identification of training loops. The number of training loops in the virtual topology influences the effectiveness of the anomaly detection scheme. In a basic-DML system employing the cross-learning mechanism, the virtual topology typically comprises one or several training loops.

C) A method for identifying abnormal training outcomes.

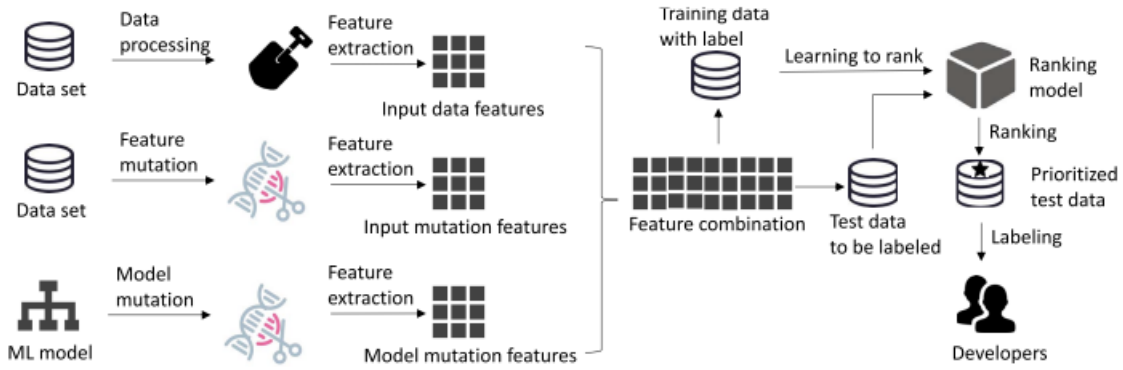
In the cross-learning mechanism, each sub-dataset is distributed to two different workers, resulting in two training results for each sub-dataset. These results are compared using the Euclidean distance to detect suspicious sub-datasets. This algorithm outlines the anomaly detection scheme, where the difference between parameter sets from the same sub-dataset is measured. If the difference exceeds a threshold ε, indicating potential poisoning, the sub-dataset is flagged and sent for relearning. However, this scheme has limitations in detecting all poisoned sub-datasets, particularly when an attacker compromises two workers with the same sub-dataset. In such cases, the center cannot differentiate between the compromised workers.

2.2 Anomaly detection using Semi Distributed Machine Learning (Semi-DML)

In this section, we introduce an enhanced anomaly detection scheme, referred to as the improved scheme, tailored for the semi-DML scenario. In semi-DML, the central entity shares spare resources for dataset training tasks. Building upon the elements of the anomaly detection scheme in the DML scenario, the improved scheme incorporates central assistance as an additional element. This allows the center to learn part or all of the sub-datasets, or verify worker results by relearning suspicious sub-datasets. The allocation of central resources significantly impacts system resource costs, making efficient resource utilization a crucial consideration.

2.3 Test prioritization using ML Prior

In this paper, we propose MLPrior, a test prioritization approach specifically designed for classical ML models. Below Fig illustrates the workflow of MLPrior.



A) Attribute Feature Generation

At the outset, MLPrior undertakes attribute feature generation by converting the attribute values of each test instance. This transformation involves converting non-numeric attributes into a numeric format. To achieve this, a mapping dictionary is created, encompassing all non-numeric attributes paired with their corresponding numeric values. For example, in the context of the attribute "gender," the values "male" and "female" are mapped to 0 and 1, respectively.

B) Mutation feature generation

MLPrior generates a collection of mutated models based on the original ML model M. MLPrior generates mutated inputs for each test instance. Subsequently, it compares the predictions of model M on the mutated input with its predictions on the original test input.

C) Feature Concatenation

For each test ML Prior concatenates the three types of feature vectors constructed in the previous steps and obtain a final vector.

D)Learning to Rank

MLPrior utilizes its Final vector as input for a pre-trained XGBoost ranking model. This model calculates the probability of the input being misclassified. Subsequently, MLPrior ranks all tests based on their probability scores in descending order, prioritizing potentially misclassified tests accordingly.

3. APPROACH

3.1 Anomaly Detection

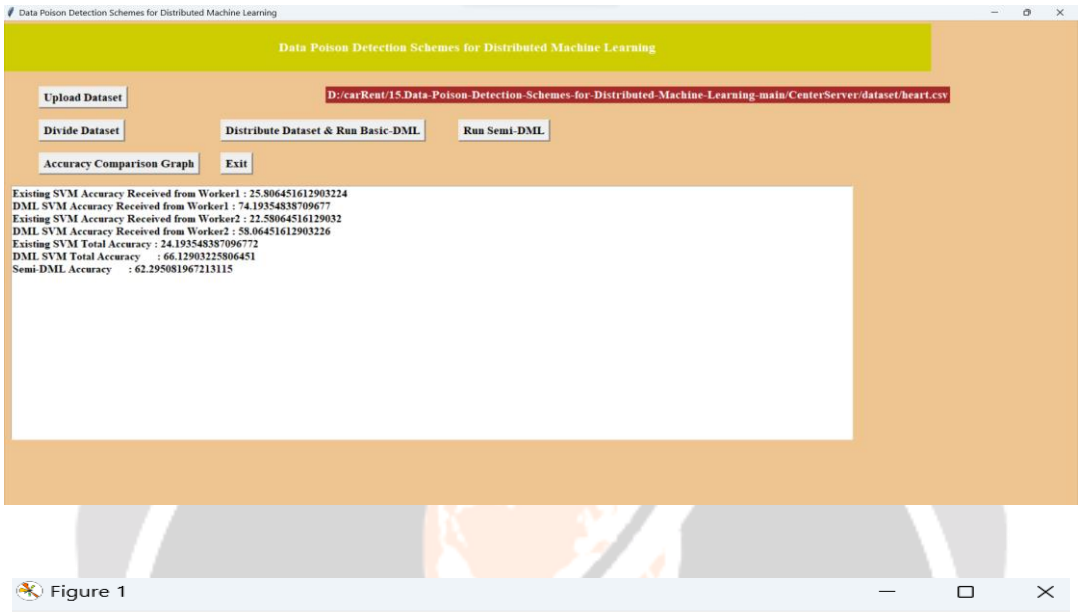
In the realm of distributed machine learning (DML), we delineate two distinct paradigms: basic-DML and semi-DML. In basic-DML, the central server delegates learning tasks to distributed machines and consolidates their learning outcomes. Conversely, in semi-DML, the central server extends its role to include direct involvement in dataset learning, augmenting its responsibilities from basic-DML. We introduce a pioneering anomaly detection scheme tailored for basic-DML, leveraging a cross-learning mechanism to uncover poisoned data instances. Demonstrating the efficacy of this mechanism, we establish its propensity to generate training loops, thereby enabling the development of a mathematical model to identify the optimal number of training loops. In the context of semi-DML, we present an enhanced anomaly detection scheme aimed at bolstering learning protection, with the central resource playing a pivotal role. To optimize system resources, we devise an approach for optimal resource allocation. Simulation results underscore the effectiveness of our proposed schemes. In the basic-DML scenario, our approach significantly enhances the accuracy of the final model, yielding improvements of up to 20% for support vector machines and 60% for logistic regression. Furthermore, in the semi-DML scenario, the improved anomaly detection scheme coupled with optimal resource allocation demonstrates a notable reduction in wasted resources, ranging from 20% to 100%. These findings underscore the efficacy and practical relevance of our proposed schemes in enhancing the security and efficiency of distributed machine learning systems.

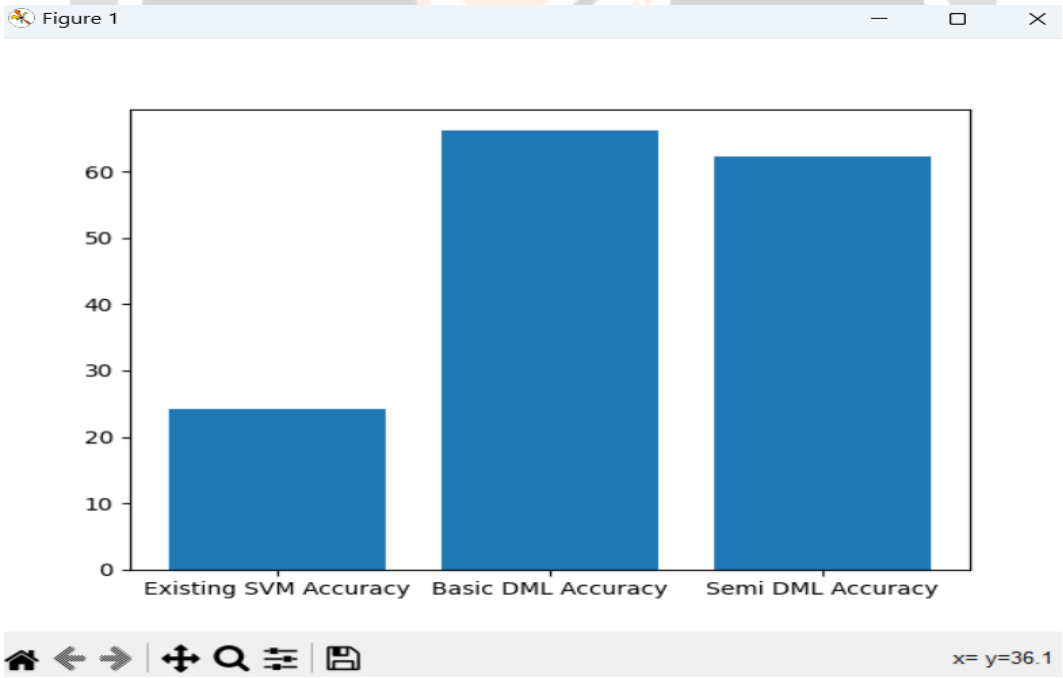
3.2 Test Prioritization

Test prioritization is done using ML Prior algorithm. MLPrior begins its process with attribute feature generation, wherein it converts the attribute values of each test instance into a numeric format. This conversion involves the creation of a mapping dictionary that pairs non-numeric attributes with their corresponding numeric values. For example, attributes such as "gender" are mapped to numeric values (e.g., "male" to 0 and "female" to 1). Following this, MLPrior proceeds to mutation feature generation, where it generates a series of mutated models based on the original ML model M. For each test instance, MLPrior generates mutated inputs, which are then compared with the predictions of model M on the original test input to identify any discrepancies. Subsequently, MLPrior conducts feature concatenation, combining the three types of feature vectors constructed in the previous steps to create a final vector for each test instance. This final vector consolidates essential features required for subsequent analysis. Finally, MLPrior employs a pre-trained XGBoost ranking model to evaluate the final vectors, calculating the probability of each input being misclassified. Based on these probability scores, MLPrior ranks all tests in descending order, prioritizing potentially misclassified tests for further investigation and corrective actions. This holistic approach ensures comprehensive detection and prioritization of potentially problematic instances within the ML model.

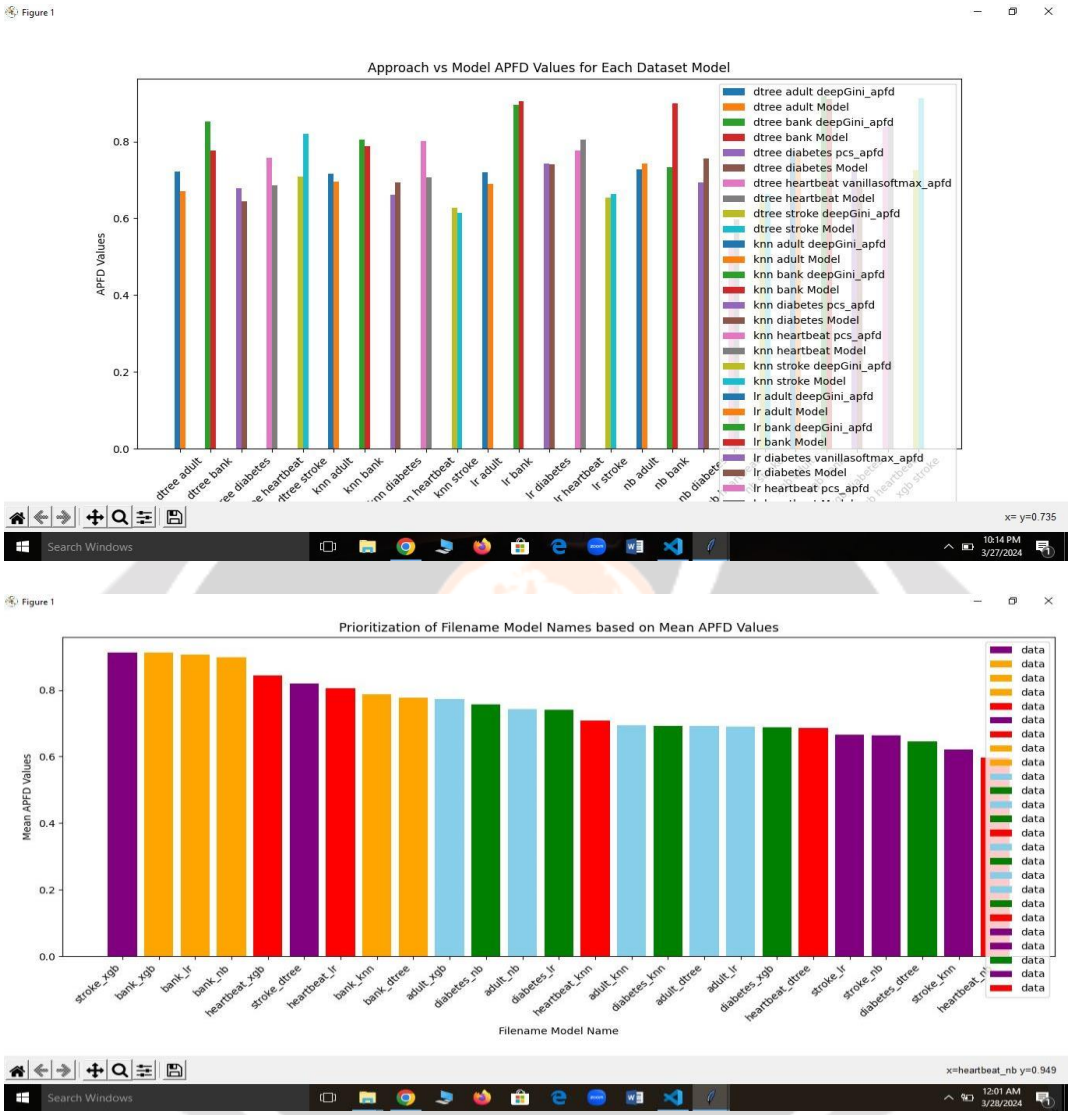
4. RESULTS

Anomaly detection where the accuracy of basic distributed machine learning and semi distributed machine learning of the data's have been shown in below figure.





Test input prioritization of multiple data for each and all models of ML Prior algorithm have been displayed in below figure.



5. CONCLUSION

The model proves the accuracy of the data for basic-DML increases up to 20% for svm and semi-DML decreases waste resource for 20-100% and the superior performance of MLPrior compared to existing methods, with an average improvement of 14.74%~66.93% on natural datasets, 18.55%~67.73% on mixed noisy datasets, and 15.34%~62.72% on fairness datasets.

6. 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] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, ‘‘Virtual adversarial training: A regularization method for supervised and semi–supervised learning,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[4] H. Smith, “Clinical AI: Opacity, accountability, responsibility and liability,” AI Soc., vol. 36, no. 2, pp. 535–545, 2021.
[5] J. Amann, A. Blasimme, E. Vayena, D. Frey, V. I. Madai, and Precise4Q Consortium, “Explainability for artificial intelligence in healthcare: A multidisciplinary perspective,” BMC Med. Inform. Decis. Making, vol. 20, pp. 1–9, Nov. 2020.
[6] T. Grote and P. Berens, “On the ethics of algorithmic decision-making in healthcare,” J. Med. Ethics, vol. 45, pp. 205–211, Nov. 2019.
[7] H. Yan et al., “New trend in Fintech: Research on artificial intelligence model interpretability in financial fields,” Open J. Appl. Sci., vol. 9, no. 10, p. 761, 2019.
[8] K. Suzuki, “Overview of deep learning in medical imaging,” Radiological Phys. Technol., vol. 10, no. 3, pp. 257–273, 2017