International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Employing Data Analytics for Identifying Potential Financial Frauds Through Adversarial Training on Imbalanced Datasets

Viketan Verma[1], Dr. Sanmati Jain[2]
Research Scholar[1], Associate Professor[2]
Vikrant University, Gwalior, India[1,2]

Abstract: Financial fraud is a major global issue, costing businesses and individuals billions of dollars annually. To combat this growing threat, organizations increasingly rely on machine learning models to detect fraudulent activities in financial transactions. However, a major obstacle in developing effective fraud detection systems is the imbalance in datasets, where genuine transactions vastly outnumber fraudulent ones. This data imbalance creates significant challenges for machine learning models, often leading to poor performance in identifying the very instances they are meant to detect. This has led to a new type of attack by users termed as adversarial machine learning in which the machine learning model used in the backend is targeted rather than the from end or the APIs. This is even more challenging due to imbalanced datasets. In conventional attacks, the first line of attack is the front end of the software. In this case, the machine learning model used in the backend is fed with bogus and/or deliberately falsified data to make it inactive. This is termed as adversarial machine learning attack or adversarial cyber-attack. It is extremely challenging to detect such attacks as there are no clear signs of attacks such as redirections, malicious code scripts, auto refresh tags etc. Instead, the data fed to the back-end machine learning model is targeted using adversarial data feeds. In this paper, a deep learning based model is used to detect such attacks which attains lower MSE compared to existing work in the domain.

Keywords: Imbalanced Datasets, Dark Web, Socio-Technical data, Adversarial Poisoning Attacks, Financial Datasets, Mean Squared Error (MSE).

I. INRRODUCTION

As the technological framework has shifted towards big data and machine learning, the types of attacks have also shifted in nature and have become much more complex in nature. A typical framework for any software utilizing machine learning is depicted in figure below [1].

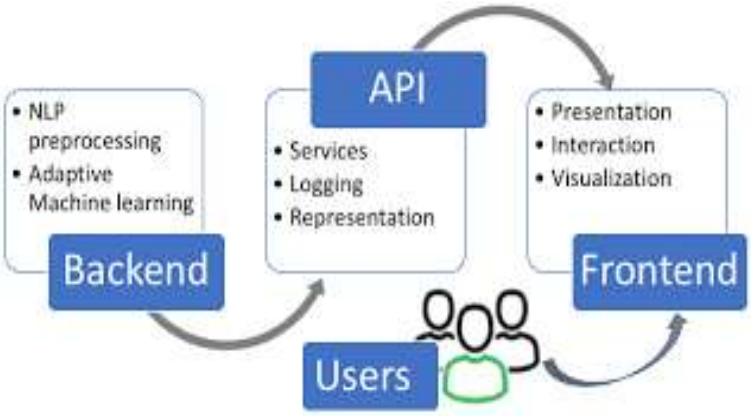

Fig.1 Machine Learning in Backend of Applications

With increasing refinement in the technological space, there lie high chances of sophisticated attacks to hinder the technological models. One such growing attack is the adversarial cyber-attack. An adversarial attack is a type of attack in which the attacker tries to feed wrong data into the training model dataset to train it to do something it must not. In short, it poisons it to learn wrong information. Today it has become a lot more sophisticated that it becomes a difficult task to detect such attacks [2].

The attacks that were prevalent mainly targeted the availability or the integrity of the machine learning models. Feeding data that renders the entire system meaningless or useless has been one of its most prominent approaches. The

consequences of such attacks are that the accuracy and performance drops drastically [3]. Also, the purpose of the entire training concept gets falsified. There are some very huge threats that these attacks pose in the form of logic corruption, data modification, data injection and transfer learning. Henceforth there is a real need of detecting and preventing adversarial attack is need of the hour. With the increase in the machine learning approaches, the adversarial attacks have also increased at a fast pace. Now, the adversaries are using very high end tools to surpass the detection mechanisms. This is very legit concern for the artificial intelligence domain to safeguard itself from such threats [4].

II. PROPOSED MODEL TO DETECT ADVERSARIAL ATTACKS

The general function of social hacking is to gain access to restricted information or to a physical space without proper permission. Most often, social hacking attacks are achieved by impersonating an individual or group who is directly or indirectly known to the victims or by representing an individual or group in a position of authority [5]. In this case, it is assumed that the data on the profiles of hackers on dark web resources can render information about future trends and aspects of cyber attacks. The mathematical model of extraction of data from dark web forums is given below [6]:

$$\boldsymbol{W(v_i, v_j) = \frac{1}{M}\sum_n \forall a, b: V(M, a)} \quad \textbf{(1)}$$

$$\boldsymbol{Or, W(v_i, v_j) = v_i^{(\beta a^{(time, M_{k,b})} - (time, M_{k,a}))} + V(M_k, b)} \quad \textbf{(2)}$$

Here,
F is a dark web forum
W is the correlation between weights
n is the number of threads analysed
v is the number of users posting messages
M is the message number/index
k is the time index
(M_k, a) & $(M_k, b)$are the messages at time index k for distinct posts a and b in the same thread K.
α, β are constants with values between 0 and 1.
v_i, v_j are distinct messages

Another approach for estimating the similarity co-efficient or the distance among the messages is given mathematically as [7]:
For two lists Γ^1 & Γ^2in the forum 'F', the similarity co-efficient or distance is computed as:

$$\boldsymbol{D^p(\Gamma^1, \Gamma^2) = \sum_{\forall i,j \in D(\Gamma^1\ \Gamma^2)} \hat{D}^p_{i,j}(\Gamma^1\ \Gamma^2)} \quad \textbf{(3)}$$

Here,
Γ^1 & Γ^2 are two lists
D^p is the distance with a penalty p
$\hat{D}^p_{i,j}$ takes up fuzzy values for different levels of similarity
(i,j) are the message pair
P is the optimistic penalty parameter
The above distance measure (Kendall's Measure) takes the relative ranking orders of any two elements in the union of two top k lists. Another measure is the absolute distance between the rankings of the same element in the union of two top k lists into consideration called the Spearman's distance measure given mathematically as [8]:

$$\boldsymbol{F^{k+1}(\Gamma^1, \Gamma^2) = \sum_{i \in D_{r1} \cap D_{r2}} |\Gamma'_1 - \Gamma'_2|} \quad \textbf{(4)}$$

Here,

F represents the Spearman's distance

D_{ri} & D_{r2} represent the domains of $\boldsymbol{\Gamma^1}$ ***and*** $\boldsymbol{\Gamma^2}$

$\boldsymbol{\Gamma'_1, \Gamma'_2}$ denoted the lists with/without entries in the original lists.

The mathematical representation of the adversarial-attack attack is given as:

Suppose the training vector be [9]:

$$\boldsymbol{Training\ Data = X(i)} \quad \textbf{(4)}$$

On Manipulation of the training vector using the poisoning vector stated by:

$$\boldsymbol{X_v = V(i)} \quad \textbf{(5)}$$

The weights of the model are controlled by the training vector and the learning algorithm that is represented mathematically as [10]:

$$\boldsymbol{w_k = f(X_v, k, f_a)} \quad \textbf{(6)}$$

Here,
$X(i)$is the real training input
$V(i)$is the poisoning vector
k is the number of iteration
fa is the activation function
$w_k$is the weight for iteration k.

The poisoning attacks very surreptitiously try to hide behind the malicious dataset and try to deceive the machine learning classifier. When the data set itself is wrong and not the intended one, it trains the neural network to do something it shouldn't. Henceforth it is kind of a very tricky mechanism being developed as adversarial attacks [11]. A very high end research is of paramount importance to able to mitigate the poisoning attacks that are prevalent. Off late there have been several cyber security practices that have come up to help the situation and deal with such sophisticated attacks. As these attacks are also of many types, a robust system is needed to thwart these types of attacks. The spamming of the data set is a popular kind of such an attack that tries to spam the training information with the adversary intended data. This can incur huge amounts of losses for the economy. Also, the wrongdoers will get a loophole to make use of to steal information and harm businesses [12]. If there isn't any system to detect the adversarial attacks then malicious software and scripts can also be injected onto the dataset that can make the neural network behave erratically and work like a malware. Henceforth a strong poisoning attack detection scheme is important to safeguard the ML classifiers from improper datasets and malicious functions [13].

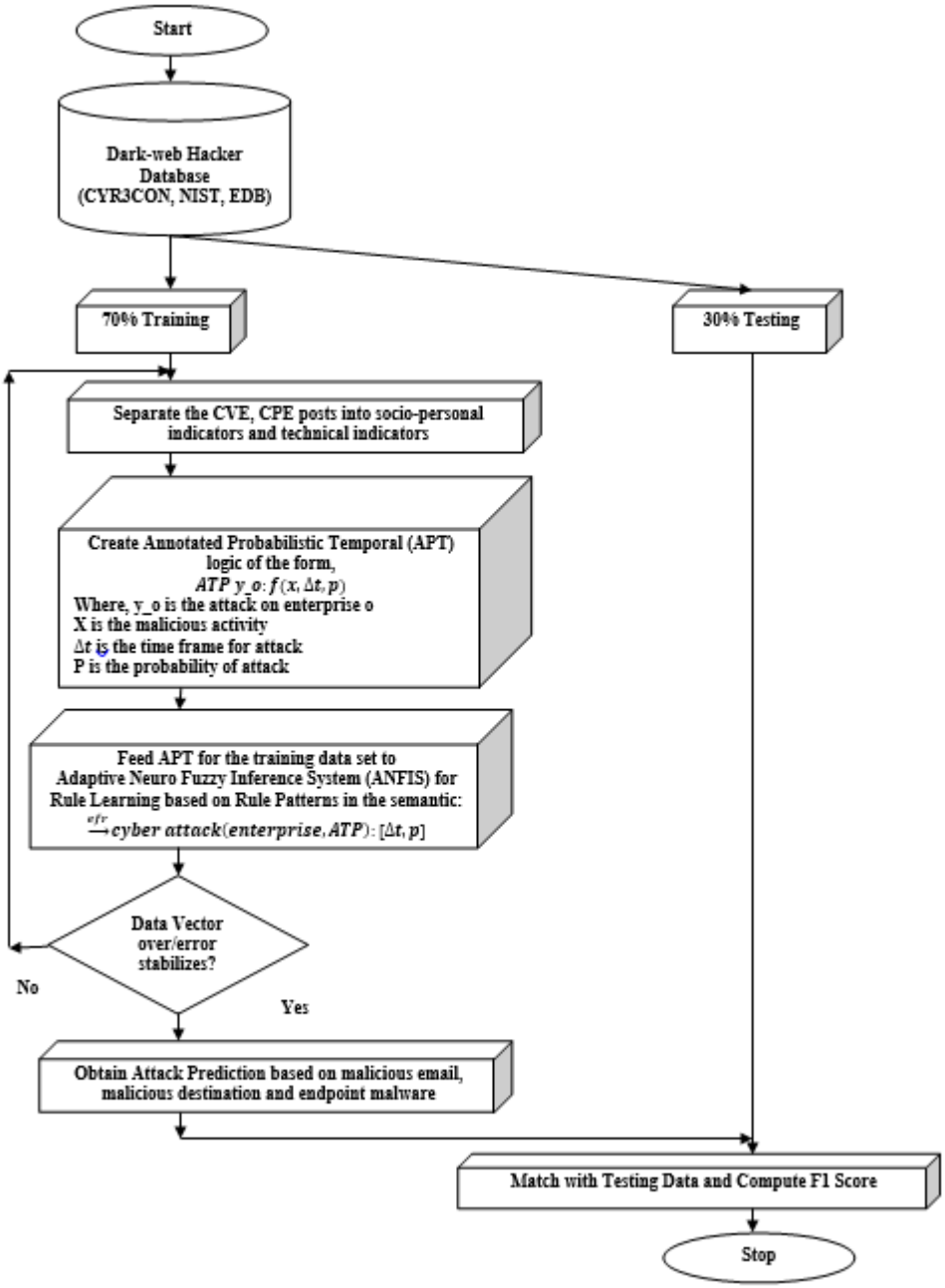

Fig.2 Proposed Flowchart

III MATHEMATICAL MODEL FOR PROPOSED WORK

Standard classifiers such as logistic regression, decision trees, and even neural networks struggle with imbalanced datasets. These models assume a relatively balanced class distribution and are biased toward the majority class. As a result, they generate high false negatives, where fraudulent transactions are misclassified as legitimate [14]. This is particularly dangerous in financial contexts, where undetected fraud can lead to significant losses, legal consequences, and reputational damage. Beyond algorithmic strategies, real-world constraints complicate fraud detection. Fraudulent patterns evolve rapidly, requiring models to adapt continuously. Labeling fraud instances accurately is difficult and often delayed, reducing the availability of real-time labeled data. Additionally, privacy concerns and regulatory requirements limit access to sensitive financial data, restricting the ability to train robust models. This creates a need for privacy-preserving learning techniques and collaboration between industry and regulatory bodies [15].

As financial institutions increasingly rely on machine learning and artificial intelligence (AI) to detect fraud, manage risks, and automate transactions, they become vulnerable to a new class of threats—adversarial poisoning attacks [16]. These attacks involve the intentional manipulation of training data to corrupt the behavior of AI models, leading them to make incorrect or biased decisions. In the context of finance, poisoning attacks can be particularly dangerous, enabling financial frauds that bypass detection systems or even exploit them to facilitate illicit activities [17]. The Back Propagation Model along with Ensemble Learning is presented din this paper. To mitigate these risks, robust learning algorithms are required—one such approach is the Quasi-Newton Backpropagation Algorithm, a powerful optimization technique capable of improving the reliability and robustness of neural networks, especially in adversarial contexts. Quasi-Newton methods are advanced optimization techniques that approximate the Hessian matrix (second-order derivatives) to improve the convergence speed and robustness of training. When integrated with backpropagation, the Quasi-Newton Backpropagation Algorithm updates weights more intelligently than simple gradient descent, adapting to the curvature of the loss function [18]. This makes it more resistant to gradient distortions caused by poisoned data and can help uncover hidden adversarial influences during the training process [19].

Start

{

Step.1: Extract dataset.

Step.2: Divide Data into training and testing samples.

Step.3: Define maximum number of iterations as $\boldsymbol{maxitr}$***.***

Step.4: Define least squares (LS) cost function to be minimized as:

$$f_{cost} = \underbrace{\boldsymbol{min}}_{maxitr} \quad \frac{1}{n}\sum_{i=1}^{n}(t_i - \hat{t}_i)^2$$

Step.5: Design a neural network and initialize weights randomly.

Step.6: for i=1:maxitr,

{

Update weights as:

$$\boldsymbol{w_{i+1} = w_i - \alpha \nabla f_{cost}(w_i) - \left[\begin{bmatrix} \frac{\partial^2 e_1}{\partial w_1^2} & \cdots & \frac{\partial^2 e_1}{\partial w_m^2} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 e_n}{\partial w_1^2} & \cdots & \frac{\partial^2 e_n}{\partial w_m^2} \end{bmatrix} * \begin{bmatrix} \frac{\partial^2 e_1}{\partial w_1^2} & \cdots & \frac{\partial^2 e_1}{\partial w_m^2} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 e_n}{\partial w_1^2} & \cdots & \frac{\partial^2 e_n}{\partial w_m^2} \end{bmatrix}^T + \alpha I\right]_i^{-1} * (t_i - \hat{t}_i)}$$

}

Step.7: if ($\boldsymbol{i == maxitr}$ ***or*** $\boldsymbol{f_{cost}}$ ***stabilizes over k-fold, validation)***

{

Truncate training

else

Update weights

}

Step.8: Computer forecasting error and accuracy at convergence.

}

Stop.

The overall performance metrics are mathematically defined as:

Accuracy: It is mathematically defined as:

$$Ac = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

Sensitivity: It is mathematically defined as:

$$Se = \frac{TP}{TP+FN} \quad (8)$$

Recall: It is mathematically defined as:

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Precision: It is mathematically defined as:

$$Precisiosn = \frac{TP}{TP+FP} \quad (10)$$

F-Measure: It is mathematically defined as:

$$F - Measure = \frac{2.Precision.Recall}{Precision+Recall} \quad (11)$$

Here.
TP represents true positive
TN represents true negative
FP represents false positive
FN represents false negative

IV EXPERIMENTAL RESULTS

The system has been designed on MATLAB. The adversarial dataset has been collected for the dark web based entries. The choice of the tool has been made based on the ease of mathematical analysis and in-built mathematical functions. The obtained results have been presented subsequently.

Suspicious String Rates	Queueing delay	Response Delay of Server	Server Response Overhead	Colliding Token Percentage	Invalid tokens percentage	Redirection Tokensper	CPU Utilization Perce
0.225179552	0.130434783	0.00210615	0.691422856	0	0.03030303	0.024096386	0.602941176
0.420363329	0.057971014	0.002737995	0.025006252	0	0.151515152	0.024096386	0.808823529
0.562737643	0.072463768	0.00463353	0.133783446	0	0.060606061	0.036144578	0.926470588
0.394169835	0.173913043	0.00716091	0.403100775	0	0.060606061	0.156626506	0.794117647
0.029995775	0.089855072	0.011162595	0.263065766	0	0	0.228915663	0.720588235
0.269961977	0.15942029	0.0084246	0.461115279	0.105263158	0	0.036144578	0.867647059
0.029995775	0.252173913	0.007728011	0.24656164	0	0	0.084337349	0.838235294
0.19391635	0.420289855	0.00926706	0.736184046	0	0	0.084337349	0.852941176
0.269961977	0.101449275	0.010741365	0.140785196	0.052631579	0	0.13253012	0.588235294
0.108998733	0.333333333	0.006139427	0.417604401	0	0	0.168674699	0.735294118
0.483734685	0.113043478	0.005370682	0.500375094	0	0	0.072289157	0.882352941
0.19391635	0.727536232	0.011162595	0.392848212	0	0	0.120481928	0.808823529
0.19391635	0.333333333	0.004422915	0.2028007	0	0.03030303	0.120481928	0.823529412
0.225179552	0.275362319	0.004991575	0.478619655	0	0.090909091	0.108433735	0.823529412
0.483734685	0.191304348	0.003053917	0.600150038	0	0.060606061	0.036144578	0.926470588
0.225179552	0.086956522	0.005370682	0.134783696	0	0.03030303	0.096385542	0.897058824
0.674271229	0.223188406	0.007476832	0.31207802	0	0.090909091	0.060240964	0.823529412
0.394169835	0.420289855	0.006002527	0.297074269	0	0.03030303	0.084337349	0.882352941
0.517955218	0.593478261	0.0042123	0.469867467	0	0	0.144578313	0.823529412
0.690747782	0.195652174	0.007476832	0.620655164	0	0.060606061	0.096385542	0.823529412
0.225179552	0.115942029	0.004001685	0.135033758	0	0	0.036144578	0.573529412
0.029995775	0.086956522	0.008319292	0.057764441	0	0	0.060240964	0.617647059
0.151246303	0.15942029	0.002611626	0.329332333	0	0	0.120481928	0.897058824
0.225179552	0.68115942	0.004317607	0.370092523	0	0	0.048192771	0.676470588
0.09252218	0.275362319	0.003053917	0.243810953	0	0.03030303	0.096385542	0.882352941
0.311364597	0.275362319	0.00547599	0.270567642	0	0.03030303	0.120481928	0.823529412
0.378115758	0.391304348	0.005139006	0.184046012	0	0.03030303	0.096385542	0.897058824

Fig.3 Raw Data

The parameters used for identifying possible attacks are:

1) Suspicious String Rates
2) Queuing Delay
3) Response Delay of the server
4) Server Response Overhead
5) Colliding Token Percentage
6) Invalid Tokens Percentage
7) Redirections Token Parentage
8) CPU utilization percentage

The total number of samples used are 128075 with 80% training and 20% testing with 25,615 samples being used for testing.

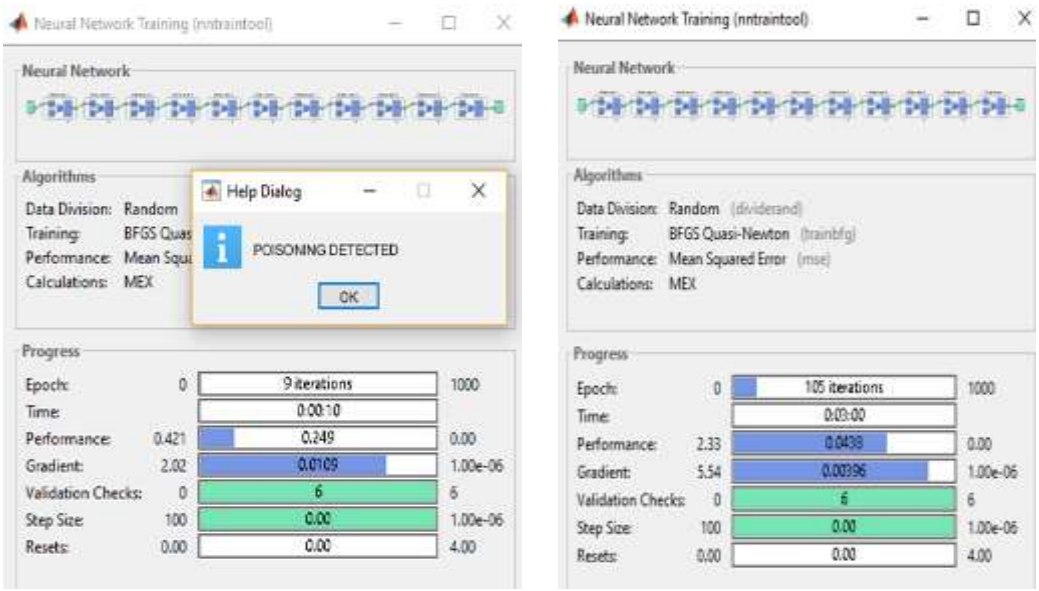

Fig.3 Training of a Neural Network Model and MSE values.

The figure depicts the training metrics for the model with the MSE value of 0.24 for the 1st dataset. Similarly, the proposed work attains an MSE of 0.043 for the next dataset

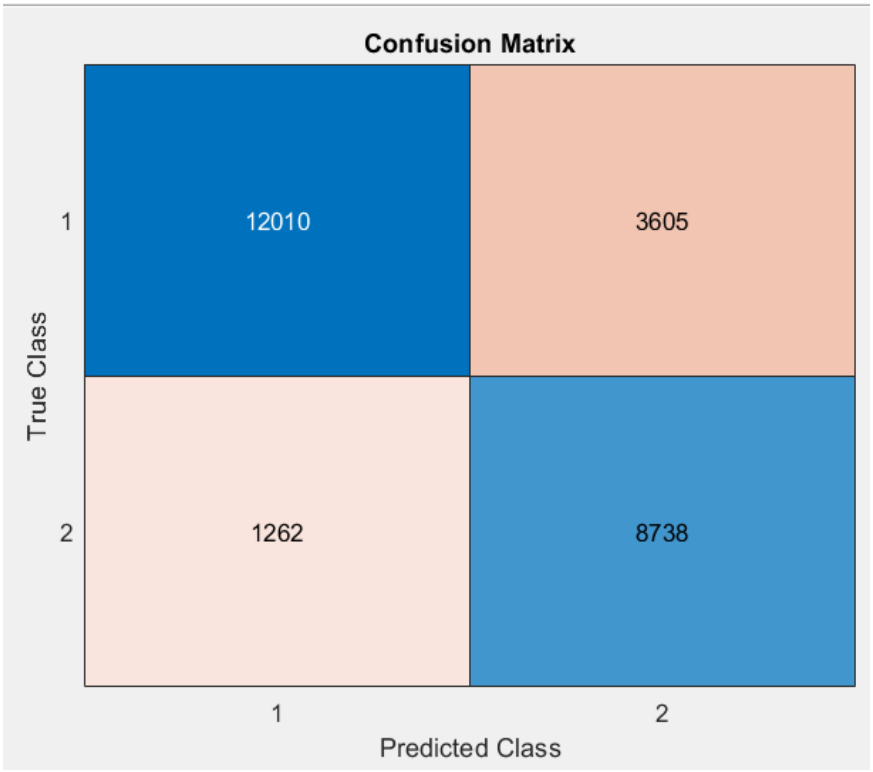

Fig.4 Confusion Matrix

The computation of the precision and accuracy can be done based on the values of TP, TN, FP and FN.

Table 1. TP, TN, FP and FN values.

S.No.	TP	TN	FP	FN
1.	12010	8738	3605	1262

The accuracy is computed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 80.99\%$$

The precision is computed as:

$$Precision = \frac{TP}{TP+FP} = 76.91\%$$

The previous work [1] attains an MSE of 0.111 and 0.070 for the 2 datasets used, but the proposed work outperforms the previous work with MSE values of 0.02 and 0.043 respectively.

Conclusion: Detecting financial adversarial poisoning attacks is exceptionally difficult because the corrupted data is often indistinguishable from legitimate data. Financial datasets are large, noisy, and subject to constant change, making it hard to spot anomalies introduced by attackers. Moreover, attackers may use slow poisoning, where small manipulations are introduced gradually, reducing the chances of detection. Even if detected, retraining models and cleaning datasets can be costly and time-consuming. It can be concluded from the previous discussions that adversarial machine learning based cyber-attacks have become a major challenge as the detection is not straightforward. With the advancement in technology, there has also been increase in cyber attacks and security breaches. Using machine learning models to predict security threats has many open research fields including predicting whether vulnerability would be exploited based on Dark Web sources. This paper presents a machine learning based approach for detecting possible adversarial attacks in advance and hence can be thwarted. The results show that the proposed work attains lower MSE compared to previous work.

References

[1] A Paudice, L Muñoz-González, A Gyorgy, EC Lupu, “Detection of adversarial training examples in poisoning attacks through anomaly detection”, IEEE Transactions on Dependable and Secure Computing, 2023, pp.1-10.

[2] G. Li, J. Wu, S. Li, W. Yang and C. Li, "Multitentacle Federated Learning Over Software-Defined Industrial Internet of Things Against Adaptive Poisoning Attacks," in IEEE Transactions on Industrial Informatics, 2022, vol. 19, no. 2, pp. 1260-1269

[3]W. Jiang, H. Li, S. Liu, X. Luo and R. Lu, "Poisoning and Evasion Attacks Against Deep Learning Algorithms in Autonomous Vehicles," in IEEE Transactions on Vehicular Technology, vol. 69, no. 4, pp. 4439-4449, April 2020

[4]E. Marin, M. Almukaynizi and P. Shakarian, "Reasoning About Future Cyber-Attacks Through Socio-Technical Hacking Information," IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 157-164.

[5] Y. Yang, L. Li, L. Chang, and T. Gu, "A Poisoning Attack Against the Recognition Model Trained by the Data Augmentation Method," in Machine Learning for Cyber Security (ML4CS 2020), Lecture Notes in Computer Science, vol. 12487, Springer, Cham, 2020, pp. 623–634.

[6] X. Zhang, X. Zhu, and L. Lessard, "Online Data Poisoning Attacks," in Proceedings of the 2nd Conference on Learning for Dynamics and Control (L4DC), 2020, pp. 201–210.

[7] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning Attack in Federated Learning using Generative Adversarial Nets," in Proceedings of the 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Rotorua, New Zealand, 2019, pp. 374–380

[8] Y. Ma, X. Zhu, and J. Hsu, "Data Poisoning against Differentially-Private Learners: Attacks and Defenses," in Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), 2019, pp. 4732–4738

[9] H. I. Kure, P. Sarkar, A. B. Ndanusa, and A. O. Nwajana, “Detecting and Preventing Data Poisoning Attacks on AI Models,” arXiv preprint, arXiv:2503.09302, 2025.

[10] A. T. Archa and K. Kartheeban, “A Review on Privacy Enhanced Distributed ML Against Poisoning Attacks,” in AI Applications in Cyber Security and Communication Networks, Lecture Notes in Networks and Systems, vol. 1032, Springer, Singapore, 2024, pp. 173–186

[11] Y. R. Maramreddy and K. Muppavaram, “Detecting and Mitigating Data Poisoning Attacks in Machine Learning: A Weighted Average Approach,” Engineering, Technology & Applied Science Research, vol. 14, no. 4, pp. 15505–15509, 2024

[12] M. Li et al., “A Sampling-Based Method for Detecting Data Poisoning Attacks in Recommendation Systems,” Mathematics, vol. 12, no. 2, p. 247, 2024.

[13] T. T. Nguyen et al., “Manipulating Recommender Systems: A Survey of Poisoning Attacks and Countermeasures,” arXiv preprint, arXiv:2404.14942, 2024

[14] M. T. Hossain, S. Islam, S. Badsha, and H. Shen, “DeSMP: Differential Privacy-exploited Stealthy Model Poisoning Attacks in Federated Learning,” arXiv preprint, arXiv:2109.09955, 2021

[15] T. Ino, K. Yoshida, H. Matsutani, and T. Fujino, “Data Poisoning Attack against Neural Network-Based On-Device Learning Anomaly Detector by Physical Attacks on Sensors,” Sensors, vol. 24, no. 19, p. 6416, 2024

[16] I. Fursov et al., “Adversarial Attacks on Deep Models for Financial Transaction Records,” arXiv preprint, arXiv:2106.08361, 2021.

[17] J. Chen, X. Zhang, R. Zhang, C. Wang, and L. Liu, “De-Pois: An Attack-Agnostic Defense against Data Poisoning Attacks,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 3412–3425, 2021

[18] Q. Zhang, F. Huang, C. Deng and H. Huang, "Faster Stochastic Quasi-Newton Methods," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 9, pp. 4388-4397, Sept. 2022

[19] D. Smyl, T. N. Tallman, D. Liu and A. Hauptmann, "An Efficient Quasi-Newton Method for Nonlinear Inverse Problems via Learned Singular Values," in IEEE Signal Processing Letters, vol. 28, pp. 748-752, 2021.