Robustness and Reliability of Machine Learning-Based Intrusion Detection Systems against Adversarial Poisoning Attacks

Sushil Buriya[1] & Neelam Sharma[2]

[1,2]Department of Computer Science, Banasthali Vidyapith, Tonk, Rajasthan, India

Abstract:- Intrusion Detection Systems (IDSs) are an essential part of security systems to protect cyber-space against the wide range of cyber-threats. In this paper, we aim to evaluate the robustness and reliability of different machine learning (ML) algorithms for IDS implementation against the feature poisoning adversarial attack and highlight their performance under adversarial and non-adversarial conditions. In order to assess the performance of ML-based IDS against poisoned training data, one benchmarked dataset NSL-KDD has been chosen. Further, Poison samples have been crafted using the Fast Gradient Sign Method (FGSM). The results show a significant decrease in accuracy and increase in false positive and false negative rate that also degrades the recall rate (RR) and precision rate (PR) along with F1-score across all the ML models when trained with poisoning datasets. The findings demonstrate that gradient boosting classifier outperforms the Logistic Regression and Multi-Layer Perceptron (MLP) algorithms with impressive results of 99.5% accuracy and 99.3% recall rate.

Keywords: *Machine Learning, Data Poisoning, Multi-Layer Perceptron, Gradient Boosting Classifier, Adversarial Attacks.*

1. Introduction

Intrusion Detection Systems (IDS) are an indispensable part of security countermeasures to defend cyber-space against the diverse range of cyber-threats [1]. IDSs are capable of analyzing the network traffic flow and system activity logs to detect the any malicious behavior of network system and network traffic. IDSs are categorized into two main types: Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDS captures incoming and outgoing network traffic and analyze the network data to find out any cyber-attack related patterns in data. Similarly, HIDS reside in end-systems of the network and monitors the user-behavior and system logs to maintain system and data integrity.

Machine learning (ML) has gained so much attraction in recent years for implementation of IDS. ML based IDS can adaptively learn from vast amounts of network and system data to detect anomalies and malicious activities in real-time. Machine learning techniques strengthen IDSs to analyze complex and dynamic network traffics by automatically identifying cyber-attack patterns and malicious activities [2]. Traditional rule-based and signature-based IDSs have limitations to synchronize with the ever-evolving nature of cyber-threats. While, ML-based IDSs have proved the impressive efficiency in detecting the novel and previously unseen attack patterns.

Moreover, supervised, unsupervised, semi-supervised, and reinforcement machine learning techniques are the various categories comes under machine learning. The mostly preferred supervised learning algorithms in intrusion detection systems (IDS) is the classification approaches to classify the system events and network traffic as malicious or benign using labeled training data. Support Vector Machines (SVM) [3], Multi-Layer Perceptron (MLP), gradient Boosting Classifier, Logistic Regression and Random Forests [4] are the supervised learning approach proposed in recent literature for implementation of ML-based IDSs in recent years. These approaches perform impressively in classification of network traffic as benign and malicious. Adversarial examples are significant threats to ML-based Intrusion Detection Systems. Adversarial examples can mislead

ML-based IDS models during both training and inference phase [5]. Data poising attacks are subset of adversarial attack to target the training phase of ML-based NIDS. Data poisoning attacks involve injecting perturbations to the training data to influence the decision making capabilities of learned model, which can be detrimental to the overall security posture of the system [6]. These adversarial attacks have been usually considered in the field of computer vision and speech recognition where an adversary generates the adversarial example and exploits the vulnerability of ML models to compromise its decision boundary to according to adversary's goal. Modern IDSs employ dynamic learning mechanisms to analyze and respond to zero day cyber-threats dynamically in real-time. Dynamic learning makes IDS to become updated with the knowledge of latest cyber-attack techniques, trends, and vulnerabilities to ensure proactive threat detection and mitigation. The nature of dynamic learning of ML-based IDS creates the opportunity for adversaries to perform the adversarial poisoning attacks [7].

The need for robust detection mechanisms to protect the ML-based IDS itself against adversarial poisoning attacks becomes prominent due to the dynamic learning and continuous learning, increasing complexity and severity of cyber-threats. There is a persistent need for robust detection mechanisms that can effectively identify and mitigate adversarial poisoning attacks to ensure the integrity and efficacy of training data for ML-based IDS in protecting against-cyber threats. The performance of ML models extremely influence by the characteristics of datasets. Some machine learning algorithms performs better on a particular type of dataset, but not on others. Poison attacks targets the learning dataset and changes the characteristics of training sets. So, the same ML model may perform differently under the poison attack circumstances. In this, study the performance of the various ML-based IDS has been assessed under the feature poisoning adversarial attack.

2. Background

Adversarial attacks against ML are classified into two main classes: data poisoning (DP) attacks and adversarial evasion attacks. Data poisoning attacks targets the learning phase, whereas adversarial evasion attacks happens during the inference or testing phase. Detailed taxonomy of these adversarial attacks illustrated in Figure-1. This paper is focused on data poisoning attacks. Data poisoning is a sophisticated cyber-attack to harm the integrity and reliable decision making of machine learning models. An adversary injects perturbed data samples during the learning phase of ML model. This attack vector exploits vulnerabilities in the machine learning algorithms involved in the model. The aim of the adversaries is to mislead the ML model through the training phase. Data poisoning attacks are implemented using several techniques and methods specially designed to target the training process and undermine the decision making capability and detection rate of machine learning model and increase the false positive and false negative rate. Data poisoning attacks typically involve injecting the malicious training samples that makes the ML classification models biased towards a target class. A ML model trained with poisoned data samples exhibits unexpected behavior during inference phase. In essence, the primary objective of the adversary is to interfere with the learning process of ML model with the intention of considerably lower the effectiveness of the target model. This interference leads to a decrease in accuracy and increase in false positives and false negative rates of the model during the testing and deployment or inference phase.

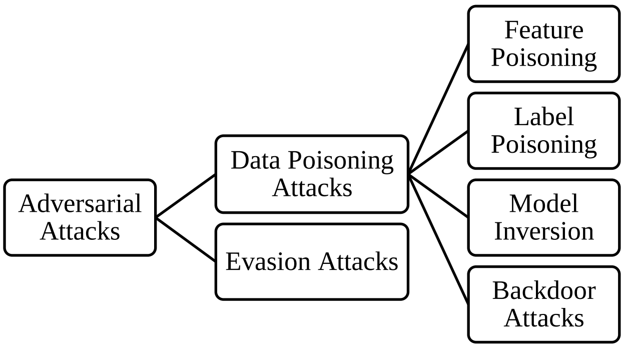

Figure 1: Taxonomy of Adversarial Attacks

The evaluation of the effect of adversarial attacks on training data samples referred to as the adversary's capacity. The adversary's primary purpose is to change either the feature values or the labels in the training set. Over the last decade, increasingly improved data poisoning algorithms have been proposed with the goal of maximizing the high rate of false positives and false negative while minimizing the number of poisoned samples necessary to carry out the attack. Adversarial poisoning attacks can take various forms to target the machine learning models that include feature poisoning [8], label poisoning [9], model inversion [10] and backdoor attacks [11], [12]. Feature poisoning attacks defines the concept of gradually modifying training data to influence model behavior. In contrast, label poisoning attacks have concept of flipping the labels or manipulating the ground truth related to training vectors. Adversary exploits vulnerabilities in the training phase of the target model to gain the insight of useful sensitive information about the training data and model learning process details in model inversion attacks. Adversary can reverse-engineer sensitive training samples or extract confidential information by analyzing the model's predictions and gradients. Adversary performs backdoor attacks by inserting backdoor patterns or triggers into the training data, which remain dormant during normal task but can be triggered to activate for specific targeted behaviors or outcomes.

Feature poisoning attacks are more practicable in ML-based IDS. An adversary alters the features or attributes of the input training vectors to compromise the performance of the IDS. Feature poison attacks targets the feature itself and are more persistent as compare to other data poison adversarial attacks. The adversary intentionally alters the features of the input training vector to introduce imperceptible perturbations that can mislead the IDS during the training process and shift its decision boundaries. These modifications are prudently crafted to avoid detection and appear legitimate. Detecting and mitigating feature poisoning attacks requires robust defense mechanisms. Data sanitization is the prominent method among defense approaches against data poisoning attacks [13]. Anomaly detection mechanism is used in data sanitation to detect adversarial examples in training data and remove it from training samples. Data sanitization techniques identify anomalies within the feature space by employing thresholding [14], nearest neighbors [15], activation space [16], or decomposing the feature covariance matrix [17]. These defensive strategies generally function under the assumption that only slight portions of the data are poisoned, thus eliminating such instances does not considerably ruin generalization. In practice, this assumption of only the small number of samples is poisoned, does not hold and easily exploited by increasing the number of adversarial examples.

In the recent research literature, numerous machine learning (ML) techniques have been proposed for intrusion detection. Typically, these algorithms are learned and tested using clean datasets. However, in this study, the primary objective is to evaluate the robustness and reliability of the ML techniques while exposed to poisoned training data and highlighting their performance under adversarial conditions. Apart from the defense mechanisms proposed in literature this study aim to find out the best performing ML algorithm for IDS under poison circumstances.

3. Methodology

In order to assess the robustness of different ML methods for IDS implementation against feature poison attack, one benchmarked dataset NSL-KDD has been chosen. Then poison samples have been crafted using the gradient based technique FGSM. Crafted adversarial samples are induced in training data for the performance evaluation of ML methods. Further explanation on the dataset, adversarial crafting and threat model and machine learning methodologies is provided in the subsequent sections.

3.1 Threat Model

Threat model is a structured representation of the possible threats and vulnerabilities for ML-based IDS. It defines the potential adversaries, their capabilities, goals, and the assets they may target within the system. A white box threat model is adopted for this study. The adversary gains comprehensive knowledge and complete access to the entire workings and behavior of the ML models in white box attacks. Adversary has sensitive information about its architecture, parameters, and decision boundaries. This level of access provides the adversary to craft highly targeted and effective adversarial examples to poison training dataset by exploiting specific vulnerabilities in the model.

The adversary aims to craft feature poisoned adversarial examples for attack instance that are identical to attack samples but classified as benign by the target IDS. Aim of adversary is to target the availability of the ML-based IDS. The adversary has knowledge and complete access to the feature space and the trained ML models. The FGSM adversarial attack is employed to craft adversarial examples by inducing the perturbation in the feature space by calculating the gradient of the loss function with respect to the input features.

3.2 Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a prevalent technique to craft adversarial examples against ML models. The FGSM attack crafts adversarial examples by calculating perturbation for input samples in such a way to maximize the loss of the learned model towards the actual output class. The goal of the FGSM attack is to craft adversarial examples that are very identical in features to one class but represents the another target class for classification problems [18].

The FGSM adversarial attack method utilizes the gradient of the target model's loss function with respect to the input training or testing samples. The gradient represents the direction of the loss function and offers important information about how small changes to the input shift the outcome of the model. Given an input sample x and its corresponding true label y_{true}, the FGSM attack calculates the gradient of the loss function with respect to the input x, loss function J, model parameter θ using the following equation:

$$D = \nabla_x(J(\theta, x, y_{true})) \tag{1}$$

Now adversarial perturbation δ is calculated by taking the sign of the gradient calculates using Equation-1 and scaling it by a small constant epsilon (ε):

$$\delta{=}\varepsilon * sign(D) \tag{2}$$

Finally, The adversarial example $x^{'}$ is crafted by adding the perturbation δ calculated using Equation-2 to actual input vector x:

$$x' = x + \delta \tag{3}$$

The epsilon (ε) parameter controls the perceptibility of the adversarial example x' by limiting the magnitude of the perturbation δ. The FGSM have capability to craft imperceptible adversarial examples towards gradient based ML models in white box adversarial threat scenario.

3.3 Machine Learning Algorithms

In order to assess the robustness of ML-based IDS against adversarial poisoning attacks various gradient based machine learning methods has been chosen. These ML approaches are widely used IDS implementations. Linear regression, logistic regression, neural network and gradient boosting machines are compared with each other for their performance on clean and poisoned data.

3.3.1 Logistic Regression

Logistic regression is a popular classification algorithm for binary classification. The core concept of this algorithm is to calculate the probability of an input sample to find out its relationship with a particular class.

Linear regression predicts continuous values, but logistic regression predicts the probability of an instance using a logistic function instead of a linear function [19].

The concept of gradients and loss function is fundamental in logistic regression in the training process. It finds the optimal weights for features that maximize the probability of the input data points. This is achieved by using a loss function that maximizes the likelihood or minimizes the negative log-likelihood. The cross-entropy loss function is employed to find out the negative log-likelihood. It measures the difference between the projected probabilities and the probability of desired class labels for the binary classification problem. The cross entropy loss is calculates as:

$$Cross\ Entropy\ Loss = \frac{1}{n}\sum_{i=1}^{n}(y_i \log{(p_i)} - (1 - y_i) \log{(1 - y_i)}) \tag{4}$$

Here n represents total number of input samples, y_i is the actual label (0 or 1) of ith input data point and p_i denotes the predicted probability that it belongs to class 1.

The logistic sigmoid function is used to calculate the probability to map the output of the linear combination of features between 0 and 1 using the following mathematical equation.

$$p = \sigma(z) = \frac{1}{1+e^{-z}} \tag{5}$$

Here *z* denotes the weighted sum of all the features x_1 to x_n with corresponding assigned weight w_0 to w_n. It is calculated as:

$$z = w_0 + w_1x_1 + w_2x_2 + \cdots \quad + w_nx_n \tag{6}$$

The weights (parameters) of the logistic regression model are updated iteratively using an optimization based on Gradient Descent during training process. The gradient of the loss function is calculated with respect to the input parameters and the parameters are adjusted in the direction of the negative gradient to minimize the loss in learning process.

3.3.2 Gradient Boosting Classifier

It is based on the concept of ensemble learning that constructs a resilient classification model by combining multiple weak learners in a sequential manner. The gradient boosting incorporates loss function to iteratively minimize the loss by adding new models to the ensemble [20]. This process effectively minimizes the loss function in the direction that decreases the errors. Initially it constructs the base model and calculates its loss value using the loss function as following:

$$F_0(x) = arg\ \underset{\gamma}{min}\frac{1}{n}\sum_{i=1}^{n} L(y_i, \gamma) \tag{7}$$

Here, $F_0(x)$ is the loss of initial base classifier and calculated with respect to predicted value γ and n is the number of input samples for training, y_i represents expected class for i*th* input vector. Loss function is mean square error that is expressed as:

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \gamma)^2 \tag{8}$$

An ensemble classifier will be learned in next step that equals to number of boosting round M. In each round pseudo residuals that is negative gradient of the loss function with respect to the prediction of the that round's classifier is calculated from m=1 to M as following:

$$r_{im} = -\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \tag{9}$$

Now, the weak classifier $h(x)$ is learned using the negative residual calculated using Equation (9) and the updated classifier in m*th* step expressed as:

$$F_m(x) = F_{m-1}(x) + \eta . arg\ \min_h \frac{1}{n} \sum_{i=1}^{n} \left(r_{im} - h(x_i) \right) \tag{10}$$

Here, η is the learning rate of boosting classifier and final boosting classifier is $F_M(x)$ after M round of boosting.

The role of gradient descent in The Gradient Boosting Classifier utilizes the gradient descent in the computation of the negative gradient or residuals in each round. The residual provides the direction of steepest descent for minimizing the loss function.

3.3.3 The Multi-Layer Perceptron

Multi-Layer Perceptron is special kind of artificial neural network architecture that contains the multiple layers of artificial neurons. It has an input layer, one or more hidden layers, and an output neuron layer [21]. Each neuron in one layer is connected to each and every another node in the succeeding layer with a weighted connection. Each layer in MLP computes the output for the subsequent layer as following:

$$z_l = \sigma_l(w_l . a_{l-1} + b_l) \tag{11}$$

Here, *w* denotes the weight matrix, *a* is the output matrix of previous layer, b denotes bias and σ represents the activation function of layer *l*. The the sigmoid (logistic), tanh, ReLU (Rectified Linear Unit), and softmax are the commonly used activation functions. ReLU (Rectified Linear Unit) activation function is used for this study. MLP uses the gradient descent with back propagation to change the weights in each round. The architecture of MLP is presented in Figure-2 with number of layers and activation functions.

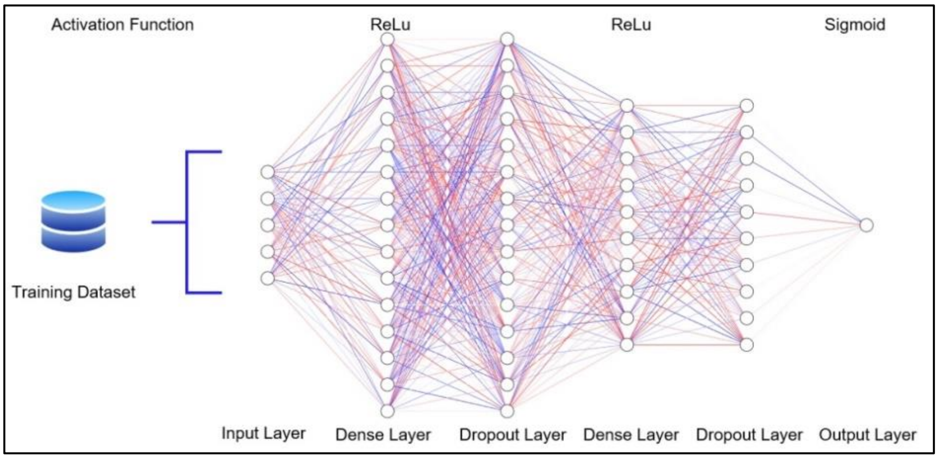

Figure 1: Graphical represenation of MPL for experimental setup

4. Experimental Setup

In the experimental setup, Machine learning algorithms gradient boosting classifier, logistic regression and multi-layer perceptron are learned using the training dataset for binary classification of the network traffic flows. NSL-KDD dataset is divided into two dataset for training and testing phase of the ML models. First all the ML algorithms are evaluated using the testing dataset using evaluation matric accuracy, recall, precision and f1-score. Further, Poison samples are crafted using FGSM adversarial crafting method using the random samples from training dataset. Poisoned samples are mixed with training data and other models are constructed using the same ML algorithms to evaluate the influence of feature poisoning attack over ML-based IDS. The experimental methodology is described in Figure-3.

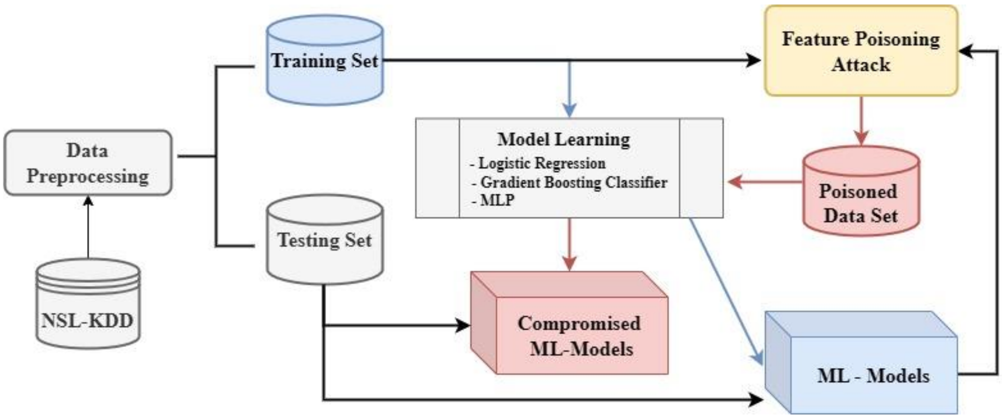

Figure 2: Proposed Experimental Methodology

4.1 Dataset Description

This study aims to assess the robustness of detection and generalization capabilities of the ML methodology for intrusion detection system under the feature poisoning attack. The experiments are conducted using an intrusion detection dataset NSL-KDD. The NSL-KDD dataset is a benchmark dataset regularly employed for assessing the performance assessment of ML-based intrusion detection systems (IDS) in recent research literature. It is an enhanced version of the original KDD Cup 1999 dataset and overcome from some of its limitations such as redundant records and biasness towards certain types of attacks. The NSL-KDD dataset covers a huge collection of network traffic flow data that includes benign and attack instances. It consists of 42 features extracted from network packets. The dataset incorporates 4 kinds of attacks including Denial of Service (DoS), Probing attacks, User-to-Root attacks (U2R) and Remote-to-Local attacks (R2L) attacks along with benign network traffic. NSL-KDD dataset provides a comprehensive and realistic environment for assessing the effectiveness of intrusion detection algorithms and systems due to its diverse range of features and attack circumstances [22].

4.2 Evaluation Metrics

Evaluation metrics are crucial for assessing the performance of machine learning tasks. In the context of binary classification of network traffic flow, positive refers to attack traffic, and negative refers to benign traffic. Accuracy, Recall Rate (RR), Precision Rate (PR), and F1-Score are used to analyze the performance of the ML models under adversarial poison attack and non-adversarial conditions. Our model predicts a binary outcome as correct (True Positive, True Negative) or incorrect prediction (False Positive, False Negative). True Positive (TP) indicates the number of attack flows properly recognized as attacks, True Negative (TN) represents the benign flows properly recognized as benign, False Positives (FP) are benign samples detected as attacks, and False Negatives (FN) are attack samples misclassified as benign. Based on these values evaluation metrics are defined.

Accuracy: It describes the frequency with which the classifier accurately identifies the true class of a given sample, whether it is an attack or benign. Specifically, this metric estimates the classifier's overall ability to correctly predict positive instances (attacks) as attacks and negative instances (benign samples) as benign. It is calculated as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{12}$$

Recall Rate: It defines the proportion of actual attack traffic samples that our model correctly identifies as attacks. Specifically, it is the percentage of attack samples in the dataset that the model successfully predicts as attacks. A high recall rate is crucial for an intrusion detection system because it indicates that the model is effective at capturing most of the attack instances.

$$Recall\ Rate\ (RR) = \frac{TP}{TP+FN} \tag{13}$$

Precision Rate: It denotes the reliability of the model's true positive predictions by calculating the percentage of properly recognized attack samples among all the samples that the model classified as attacks. High precision means that most of the samples predicted as attacks are indeed attacks, reflecting the model's effectiveness in minimizing false positives.

$$Precision\ Rate\ (PR) = \frac{TP}{TP+FP} \tag{14}$$

F1 Score: The F1 Score represents the collective measure of precision rate and recall rate that provides a balance in measure of a model's performance in binary classification. It is considered as the harmonic mean of both precision rate and recall rate.

$$F1\ Score\ (F1) = 2 * \frac{PR*RR}{PR+RR} \tag{15}$$

5. Results and Discussions

The network traffic data from the NSL-KDD dataset is randomly split into two separate datasets for the purpose of training and testing of ML models in which 80% of the network traffic samples used for training and 20% for testing purpose. The performance of ML algorithms is assessed with evaluation metrics accuracy, recall, precision, and F1-score. Initially, the performance assessment of ML models is conducted using uncontaminated datasets for both the training and testing phase. Subsequently, the performance of the ML models is assessed using a poisoned training set which contains both clean samples and adversarial samples crafted using the feature poisoning attack and keeping the testing set clean. The testing set is identical for both evaluations but the models are trained using either a poisoned or a clean dataset.

Table 1: Performance Assessment of ML-based IDS with clean datasets

ML Models	Accuracy	Recall Rate	Precision Rate	F1-score
Logistic Regression	**0.86**	**0.841**	**0.86**	**0.85**
Gradient Boosting Classifier	**0.995**	**0.993**	**0.996**	**0.995**
MLP	**0.974**	**0.977**	**0.969**	**0.973**

Tables 1 and 2 demonstrate the results achieved with clean samples in both training and testing sets and using poisoned samples in the training set and clean samples in the testing set, respectively. The findings show a decrease in accuracy, recall, precision, and F1-score across all the ML models when trained with poisoned datasets.

Table 2: Performance Assessment of ML-based IDS under feature poison attack

ML Models	Accuracy	Recall Rate	Precision Rate	F1-score
Logistic Regression	**0.52**	**0.53**	**0.4**	**0.45**
Gradient Boosting Classifier	**0.71**	**0.66**	**0.8**	**0.72**
MLP	**0.65**	**0.63**	**0.68**	**0.65**

Gradient boosting classifier outperforms the logistic regression and multi-layer perceptron other ML algorithms with impressive results of 99.5% accuracy and 99.3% recall. The performance of MLP is somewhat less as compare to gradient boosting classifier. Logistic regression is not a good choice for implementation for IDS.

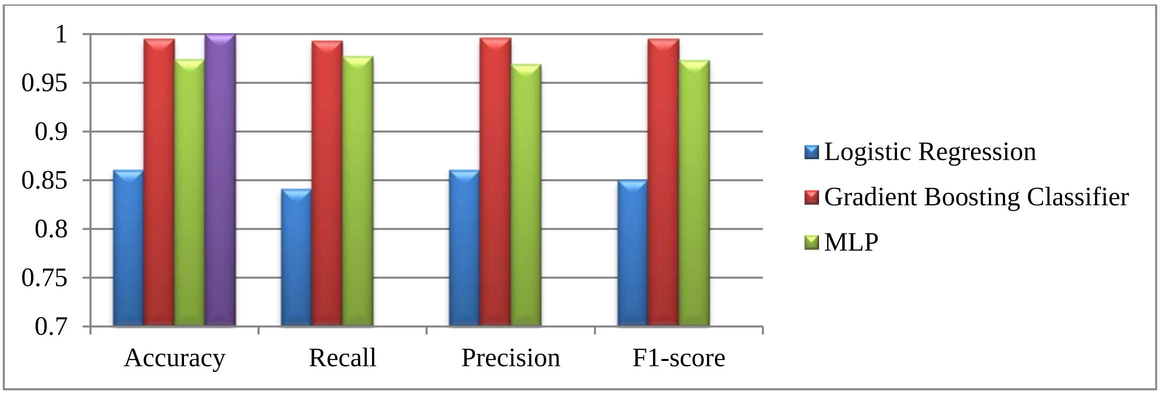

Figure 3: Performance Assessment of ML-based IDS with clean datasets

Accuracy of logistic regression, gradient boosting classifier and MLP has been decreased and affected under the influence of adversarial poison attack. Feature poisoning attacks equally affects the precision and recall in MLP, but recall is more affected in gradient boosting classifier. The rate of false negative is increased as compare to false positive in Gradient boosting classifier under the adversarial poisoning attack.

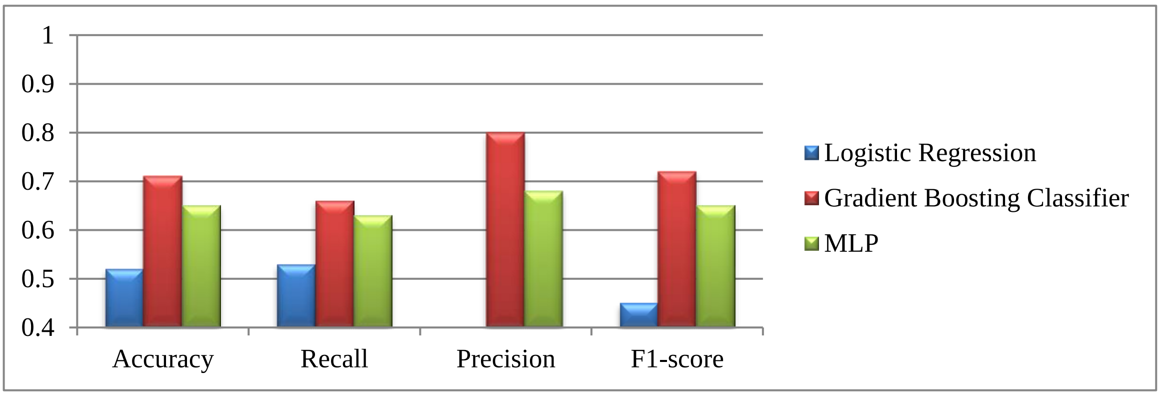

Figure 4: Performance Assessment of ML-based IDS with poisoned training set

6. Conclusion

The performance of the various ML-based IDS has been assessed under the feature poisoning adversarial attack. The results demonstrate a significant decline in accuracy, recall, precision, and F1-score for assessed models learned logistic regression, gradient boosting classifier, and multi-layer perceptron under feature poisoning adversarial attacks. Gradient boosting classifier outperformed the others, achieving 99.5% accuracy and 99.3% recall with clean dataset. The study demonstrates the critical need for robust defense mechanisms against data poisoning attacks for ML-based Intrusion Detection Systems (IDS). These attacks compromise the training datasets to interfere in the learning phase of the target models and compromise the availability of ML-based IDS for true network traffic. The performance evaluation is performed with the NSL-KDD dataset and insights the vulnerability of ML models against adversarial attacks and emphasizing the importance of enhancing IDS resilience to maintain cyber security.

Refrences

[1] J. Du, K. Yang, Y. Hu and L. Jiang, "*NIDS-CNNLSTM: Network Intrusion Detection Classification Model Based on Deep Learning,*" in IEEE Access, vol. 11, pp. 24808-24821, 2023, doi: 10.1109/ACCESS.2023.3254915.

[2] G. Abdelmoumin, D. B. Rawat and A. Rahman, "*On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Systems for the Internet of Things,*" in IEEE Internet of Things Journal, vol. 9, no. 6, pp. 4280-4290, 15 March15, 2022, doi: 10.1109/JIOT.2021.3103829.

[3] M. V. Kotpalliwar and R. Wajgi, "*Classification of Attacks Using Support Vector Machine (SVM) on KDDCUP'99 IDS Database,*" 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 2015, pp. 987-990, doi: 10.1109/CSNT.2015.185.

[4] Wu, T., Fan, H., Zhu, H. et al. "*Intrusion detection system combined enhanced random forest with SMOTE algorithm.*" in EURASIP J. Adv. Signal Process. 2022, 39 (2022). https://doi.org/10.1186/s13634-022-00871-6.

[5] J. Liu, M. Nogueira, J. Fernandes and B. Kantarci, "*Adversarial Machine Learning: A Multilayer Review of the State-of-the-Art and Challenges for Wireless and Mobile Systems,*" in IEEE Communications Surveys & Tutorials, vol. 24, no. 1, pp. 123-159, Firstquarter 2022, doi: 10.1109/COMST.2021.3136132.

[6] L. Sun, M. Tan, and Z. Zhou, "*A survey of practical adversarial example attacks," Cybersecurity"*, vol. 1, no. 1, p. 9, Sep. 2018, doi: 10.1186/s42400-018-0012-9.

[7] X. Yuan, P. He, Q. Zhu and X. Li, *"Adversarial Examples: Attacks and Defenses for Deep Learning,"* in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 9, pp. 2805-2824, Sept. 2019, doi: 10.1109/TNNLS.2018.2886017.

[8] A. Shafahi, W. Ronny Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, *"Poison frogs! targeted clean-label poisoning attacks on neural networks."* in Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18), 2018, doi: 10.48550/arXiv.1804.00792.

[9] B. Biggio, B. Nelson, and P. Laskov, *"Poisoning Attacks against Support Vector Machines."* arXiv, Mar. 25, 2013. doi: 10.48550/arXiv.1206.6389.

[10] *S. Zhou, T. Zhu, D. Ye, X. Yu and W. Zhou, "Boosting Model Inversion Attacks With Adversarial Examples," in IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 3, pp. 1451-1468, May-June 2024, doi: 10.1109/TDSC.2023.3285015.*

[11] K. Doan, Y. Lao, W. Zhao, and P. Li, "LIRA: Learnable, Imperceptible and Robust Backdoor Attacks," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 11946–11956. doi: 10.1109/ICCV48922.2021.01175.

[12] Khoa Doan, Yingjie Lao, Weijie Zhao and Ping Li, "Lira: Learnable imperceptible and robust backdoor attacks", Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11966-11976, 2021, doi: 10.24963/ijcai.2022/239.

[13] S. Ho, A. Reddy, S. Venkatesan, R. Izmailov, R. Chadha, and A. Oprea, "Data Sanitization Approach to Mitigate Clean-Label Attacks Against Malware Detection Systems," in *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, Nov. 2022, pp. 993–998. doi: 10.1109/MILCOM55135.2022.10017768.

[14] J. Steinhardt, P. W. Koh, and P. Liang, "Certified Defenses for Data Poisoning Attacks." arXiv, Nov. 23, 2017. doi: 10.48550/arXiv.1706.03691.

[15] N. Peri *et al.*, "Deep k-NN Defense against Clean-label Data Poisoning Attacks." arXiv, Aug. 13, 2020. doi: 10.48550/arXiv.1909.13374.

[16] B. Chen *et al.*, "Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering." arXiv, Nov. 08, 2018. doi: 10.48550/arXiv.1811.03728.

[17] B. Tran, J. Li, and A. Madry, "Spectral Signatures in Backdoor Attacks." arXiv, Nov. 01, 2018. doi: 10.48550/arXiv.1811.00636.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples." arXiv, Mar. 20, 2015. doi: 10.48550/arXiv.1412.6572.

[19] B. Subba, S. Biswas, and S. Karmakar, "Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression," in *2015 Annual IEEE India Conference (INDICON)*, Dec. 2015, pp. 1–6. doi: 10.1109/INDICON.2015.7443533.

[20] M. Madhavi and N. P. Nethravathi, "Intrusion Detection in Networks using Gradient Boosting," in *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, Apr. 2023, pp. 139–145. doi: 10.1109/ICAECIS58353.2023.10170199.

[21] P. Shettar, A. V. Kachavimath, M. M. Mulla, N. D. G, and G. Hanchinmani, "Intrusion Detection System using MLP and Chaotic Neural Networks," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2021, pp. 1–4. doi: 10.1109/ICCCI50826.2021.9457024.

[22] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.