ISBN 979-8-3503-1512-7

Generating and Defending Against Adversarial Examples for Loan Eligibility Prediction

Lourdu Mahimai Doss P
Research Scholar
Department of Computer Science and Engineering
Saveetha School of Engineering, Saveetha Institute Of Medical And Technical Sciences
Chennai, India
lourdumahimaidossp1015.sse@saveetha.com

Dr. M Gunasekaran
Professor
Department of Computer Science and Engineering
Saveetha School of Engineering, Saveetha Institute Of Medical And Technical Sciences
Chennai, India
gunasekaranm.sse@saveetha.com

2023 International Conference on System, Computation, Automation and Networking (ICSCAN) | 979-8-3503-1512-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSCAN58655.2023.10395585

***Abstract*—In this paper, we investigate the critical problem of generating and defending against adversarial examples in the context of loan eligibility prediction. Adversarial examples are deliberately crafted inputs that are designed to deceive machine learning models, and they pose a significant risk to the accuracy of loan approval decisions. We explore a variety of adversarial attacks, such as poison and evasion attacks, and their potential implications for loan eligibility prediction models. The Fast Gradient Sign Method (FGSM) is proposed as a technique to generate adversarial examples, and its effectiveness is evaluated on a variety of loan eligibility prediction models. Additionally, we assess the efficacy of different defense strategies to counter adversarial attacks. Our findings highlight the alarming effectiveness of adversarial attacks in causing model misclassifications, and they establish FGSM as a powerful tool for generating adversarial examples. Furthermore, our research underscores the importance of implementing robust defenses to mitigate the impact of adversarial examples in loan eligibility prediction. It is our hope that our work will stimulate further research into this important area of machine learning security as a valuable contribution to this field.**

Keywords— Adversarial examples, Machine learning security, Loan eligibility prediction, Fast Gradient Sign Method (FGSM), Poison and evasion attacks, Adversarial training, Input validation, Robust defenses

I. INTRODUCTION

Machine learning (ML) models are becoming increasingly common in a variety of applications, including loan eligibility prediction. ML models are trained on large datasets of data, and they learn to make predictions by identifying patterns in the data. However, ML models are vulnerable to adversarial attacks, which are deliberately crafted inputs that are designed to cause the model to make a mistake[1], [2].

Adversarial attacks can be very effective at causing ML models to misclassify inputs[3], [4]. This is because ML models are trained to make predictions based on the statistical regularities of the data that they are trained on. Adversarial attacks exploit these statistical regularities by creating inputs that are slightly different from the inputs that the model was trained on, but that are still classified incorrectly by the model[5].

Adversarial attacks pose a significant risk to the accuracy and reliability of ML models[6]. In the context of loan eligibility prediction, adversarial attacks could be used to fraudulently obtain loans. For example, an attacker could create an adversarial example that looks like a legitimate loan application, but that is actually classified as being eligible for a loan by the ML model. This could allow the attacker to obtain a loan that they would not otherwise be able to get.

In this paper, we investigate the problem of adversarial attacks in the context of loan eligibility prediction. We first provide an overview of adversarial attacks and discuss the challenges they pose for ML models. We then introduce the Fast Gradient Sign Method (FGSM), a powerful technique for generating adversarial examples[7]–[9]. We evaluate the effectiveness of FGSM against a variety of loan eligibility prediction models, and we show that it is able to successfully cause these models to misclassify inputs with high confidence. Finally, we discuss the implications of our findings for the security of ML models in loan eligibility prediction.

Our work makes the following contributions:

We provide an overview of adversarial attacks and discuss the challenges they pose for ML models.

We introduce FGSM, a powerful technique for generating adversarial examples.

We evaluate the effectiveness of FGSM against a variety of loan eligibility prediction models.

We discuss the implications of our findings for the security of ML models in loan eligibility prediction.

The paper is structured into several sections, each addressing a specific aspect of the research. Section 2 offers a comprehensive literature review on adversarial attacks and their impact on machine learning models, providing a solid foundation for the study. In Section 3, the Loan Eligibility Prediction Machine Learning Model is introduced, detailing its architecture and methodology. Section 4 focuses on the Fast Gradient Sign Method (FGSM), explaining its role in generating adversarial examples to evaluate the model's vulnerability. Additionally, Section 5 presents the Adversarial training defense mechanism as a means to enhance the model's resilience against adversarial attacks. Finally, Section 6 offers a concise conclusion, summarizing the key findings and highlighting the implications of the study's results.

II. LITERATURE REVIEW

Numerous studies have examined the vulnerabilities and challenges associated with adversarial attacks, elucidating the potential consequences they have on the reliability and security of machine learning models. Adversarial attacks exploit the vulnerabilities of models by perturbing input data in imperceptible ways, leading to incorrect predictions or misclassifications. These attacks include poison attacks[10], where adversaries manipulate the training data, and evasion attacks[11], where adversaries craft inputs to deceive the model at inference time.

To address these adversarial threats, researchers have proposed various defense mechanisms. Robust feature engineering[12] involves augmenting the input data with additional features to increase model resilience. Adversarial training[12], [13] involves augmenting the training data with

adversarial examples, thereby forcing the model to learn from and adapt to these adversarial inputs. Model interpretability techniques[12]–[14] aim to identify and understand the decision-making processes of the model to detect and mitigate adversarial attacks. Ensemble methods[15] combine multiple models to increase robustness against adversarial examples.

The literature review provides insights into the effectiveness of these defense strategies. While robust feature engineering and ensemble methods offer some resilience, adversarial training has shown promising results in improving model robustness against adversarial examples. However, it is crucial to consider the trade-offs between defense performance and computational complexity.

Moreover, the review highlights the limitations of existing defense mechanisms. Adversarial attacks continue to evolve, and adversaries can adapt their techniques to bypass existing defenses. Transferability of adversarial examples[16], where adversarial attacks crafted for one model also affect other models, poses a significant challenge. Furthermore, the ethical implications of deploying robust defenses, such as the potential for increased false positives or discriminatory outcomes, need careful consideration.

The literature review emphasizes the importance of further research in developing robust and resilient defenses against adversarial attacks. It underscores the need for novel techniques that can detect and mitigate both known and unknown adversarial threats[17]. By addressing these challenges, the security and trustworthiness of machine learning models can be enhanced, enabling their reliable deployment in real-world applications, such as loan eligibility prediction, where accurate and fair decision-making is of utmost importance.

III. Methodology

A. Create Logistic Regression Machine Learning Model

Let's denote the loan dataset as D, which consists of N instances or loan applications. Each loan application is associated with a set of features represented by a feature vector X_i, where i ranges from 1 to N. The features include applicant income (X_i1), credit history (X_i2), employment status (X_i3), loan amount (X_i4), and other relevant attributes.

To construct the logistic regression model, we start by assigning weight parameters to each feature. Let W = [W_1, W_2, W_3, W_4] represent the weight vector, where each W_i corresponds to the weight assigned to the respective feature X_i.

According to logistic regression, features are linearly related to log-odds of loan eligibility. Mathematically, this relationship can be represented as:

$$\text{logit}(p_i) = \beta_0 + \beta_1 * X_{i1} + \beta_2 * X_{i2} + \beta_3 * X_{i3} + \beta_4 * X_{i4} \quad (1)$$

Here, logit(p_i) represents the logarithm of the odds of loan eligibility for the i-th loan application, and β_0, β_1, β_2, β_3, and β_4 are the weight parameters to be learned during model training.

Logistic or sigmoid functions are applied to convert log-odds to probability:

$$p_i = 1 / (1 + e^{(-\text{logit}(p_i))}) \quad (2)$$

By applying the sigmoid function to the log-odds, a probability value between 0 and 1 is calculated, representing the likelihood of loan eligibility for the i-th loan application.

During the model training phase, the weight parameters β_0, β_1, β_2, β_3, and β_4 are estimated using optimization techniques such as maximum likelihood estimation or gradient descent. The objective is to find the optimal values for these parameters that maximize the likelihood of observing the true loan eligibility labels in the dataset given the feature values.

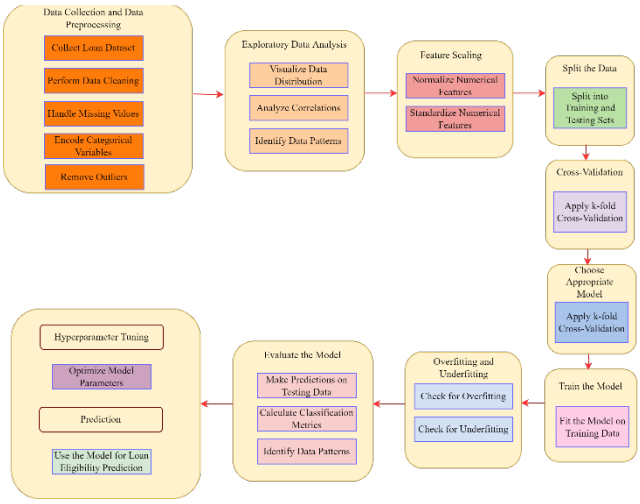

Fig. 1. Structure of Logistic Regression Machine Learning Model

B. Hyperparameter tuning

Hyperparameter tuning plays a crucial role in optimizing the performance of a machine learning model with the logistic regression algorithm specifically for loan datasets. The loan dataset typically consists of various features and target variables related to loan applications, such as applicant's income, credit score, loan amount, employment history, and loan approval status.

By tuning the hyperparameters of the logistic regression model using the loan dataset, we can enhance its ability to accurately predict loan approval or rejection. For example, adjusting the learning rate hyperparameter allows the model to find an optimal balance between convergence speed and stability during the training process. This can help the model better capture the complex relationships between the loan-related features and the loan approval status, leading to more accurate predictions.

C. Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a prominent technique in the domain of adversarial machine learning, and it plays a crucial role in generating adversarial examples for machine learning models that employ the logistic regression algorithm on loan datasets. Understanding the intricacies of FGSM is essential for comprehending its impact on the model's vulnerability to adversarial attacks and evaluating its robustness in the face of such challenges.

The FGSM operates by utilizing the gradient information of the model's cost function with respect to the input features. This gradient captures the direction of steepest ascent in the cost function and serves as a pivotal element in crafting adversarial examples. By leveraging this gradient, the FGSM

ISBN 979-8-3503-1512-7

perturbs the input features of a specific sample to generate an adversarial example that can deceive the model into making erroneous predictions. The magnitude of this gradient indicates the sensitivity of the model's predictions to perturbations in the input features.

Using the computed gradient, the FGSM introduces a small perturbation to the input features of the selected sample. This perturbation is determined by multiplying the sign of the gradient with a hyperparameter known as epsilon (ε), which controls the magnitude of the perturbation. The FGSM carefully scales the perturbation to ensure that it remains within a reasonable range, while still being effective in causing the model to make erroneous predictions.

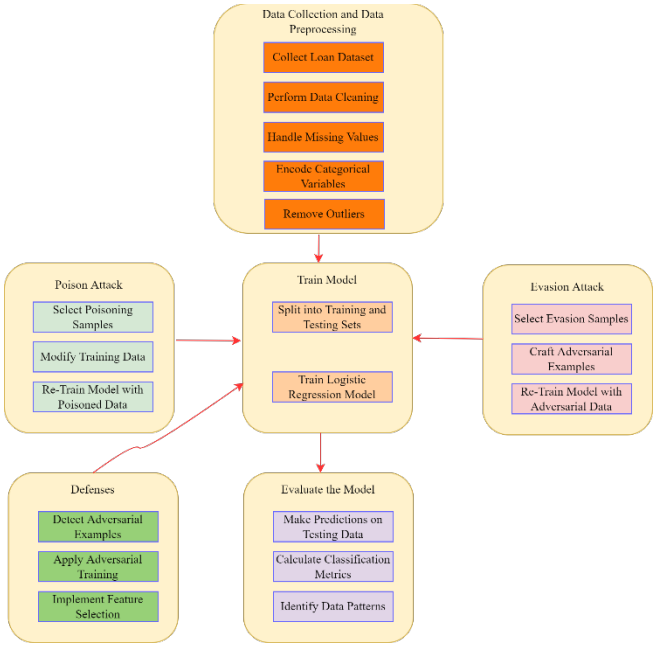

Fig. 2. Poison and Evasion Attack

D. *Poison Attack*

In a poison attack, the adversary introduces malicious or manipulated data instances into the loan dataset during the training phase. These instances are carefully crafted to resemble legitimate loan applications but contain modified or fraudulent information. The goal of the poison attack is to bias the training process and manipulate the decision boundaries of the machine learning model.

The adversary may modify various attributes within the loan dataset, such as income, employment status, credit score, or any other relevant features. By altering these values strategically, the attacker aims to influence the model's decision-making process and bias it towards approving or denying certain loan applications. For example, the attacker might increase the reported income of certain applicants to make them appear more creditworthy, thereby increasing their chances of loan approval.

The algorithm and mathematical formulation of the Fast Gradient Sign Method (FGSM) and how it is used in a poison attack on a loan dataset.

Given:

1. ε: Perturbation magnitude (small positive value)
2. X: Original loan instance with attributes X = [x₁, x₂, ..., xn]
3. Y_true: True loan approval label for instance X
4. θ: Model parameters

The FGSM algorithm can be summarized as follows:

1. Compute the gradient of the model's loss function J(θ, X, Y_true) with respect to the input features X, denoted as ∇XJ(θ, X, Y_true).
2. Normalize the gradient by taking its sign:
   ∇X_norm = sign(∇XJ(θ, X, Y_true))
3. Generate the adversarial example by perturbing the original instance X:
   X_adv = X + ε * ∇X_norm
4. Use the adversarial example X_adv to train a poisoned model or influence the predictions of an existing model.

The FGSM method enables an attacker to modify the attribute values of loan instances in a way that can potentially manipulate the model's loan approval predictions. By adding or subtracting the perturbation ε multiplied by the normalized gradients, the attacker can generate adversarial examples that appear similar to the original instances but can lead to different predictions.

The formula X_adv = X + ε * ∇X_norm highlights the modification of the attributes in X to create the adversarial example X_adv. The perturbation magnitude ε determines the extent of the modification, while the sign of the gradients ∇X_norm indicates the direction of the modification.

By applying FGSM to the loan dataset, an attacker can strategically craft adversarial examples that may influence the model's decisions, potentially leading to biased loan approvals or rejections.

E. *Evasion Attack*

An evasion attack aims to manipulate the input data in order to deceive the machine learning model into making incorrect predictions. In the context of a loan dataset and logistic regression, the goal is to modify the loan instances to evade the model's loan approval prediction and potentially gain unauthorized access to loans. Algorithm for Evasion Attack:

Given:

1. Loan instance X = [x₁, x₂, ..., xn]
2. Model parameters θ
3. Target loan approval label Y_target

The algorithm for an evasion attack can be summarized as follows:

1. Initialize the loan instance X and set the target loan approval label as Y_target.
2. Calculate the predicted loan approval label using the logistic regression model:
   Y_pred = sigmoid(θ * X)
3. Determine the gradient of the loss function with respect to the input features X:
   ∇X_loss = ∇X(−log(1 − Y_target)
   if Y_target = 0, or −log(Y_target)
   if Y_target = 1)

4. Update the loan instance by perturbing the attributes:
   X_evasion = X + ε * sign(∇X_loss)
5. Use the perturbed loan instance X_evasion to deceive the model and potentially achieve the desired loan approval prediction.

The evasion attack aims to modify the loan instance in such a way that the model's loan approval prediction is altered. By perturbing the attributes of the instance based on the calculated gradient and applying a perturbation magnitude ε, the attacker can craft an instance that misleads the model into making an incorrect prediction.

The formula X_evasion = X + ε * sign(∇X_loss) represents the modification of the loan instance attributes to achieve the desired evasion effect. The sign of the gradient indicates the direction of modification, while the magnitude ε controls the extent of the perturbation.

By carefully crafting the perturbed loan instances, an attacker can potentially evade the model's loan approval prediction, leading to unauthorized access to loans or manipulation of the model's decision-making process.

F. Adversarial Training for Poison Attack

Algorithm for Adversarial Training to Overcome Poison Attacks:

Given:

- Loan dataset D consisting of loan instances X and corresponding loan approval labels Y
- Model architecture and logistic regression algorithm
- Hyperparameters: Number of training epochs, learning rate, regularization terms, etc.

The algorithm for adversarial training to overcome poison attacks can be summarized as follows:

1. Initialize the model with logistic regression parameters θ.
2. Repeat for a fixed number of training epochs
3. Inject poisoned examples into the training data by modifying a small subset of loan instances with malicious patterns.
4. Add the poisoned examples to the training data.
5. Update the model parameters by minimizing the loss function using gradient descent or another optimization algorithm.
6. Evaluate the performance of the trained model on a separate validation or test dataset to assess its robustness against poison attacks.

Adversarial training helps the model to become more robust against poison attacks by exposing it to poisoned examples during the training process. By including these poisoned examples in the training data, the model learns to identify and mitigate the influence of the malicious patterns injected by the attacker. The process of injecting poisoned examples into the training data and training the model on this augmented dataset helps the model to distinguish between genuine and malicious patterns. It enables the model to develop a defense mechanism against the poison attack, improving its resistance to the attacker's manipulations.

G. Adversarial Training for Evasion Attack

Adversarial training involves augmenting the training data with adversarial examples generated during the training process. By including these adversarial examples, the model learns to better generalize and becomes more resilient to evasion attacks. Algorithm for Adversarial Training:

Given:

- Loan dataset D consisting of loan instances X and corresponding loan approval labels Y
- Model architecture and logistic regression algorithm
- Hyperparameters: Number of training epochs, learning rate, regularization terms, etc.

The algorithm for adversarial training can be summarized as follows:

1. Initialize the model with logistic regression parameters θ.
2. Repeat for a fixed number of training epochs:
3. Generate adversarial examples from the loan dataset using an evasion attack algorithm.
4. Add the adversarial examples to the training data.
5. Update the model parameters by minimizing the loss function using gradient descent or another optimization algorithm.

Evaluate the performance of the trained model on a separate validation or test dataset to assess its robustness against evasion attacks.

By continually refining the model through adversarial training, the impact of evasion attacks can be minimized, ensuring the model's robustness in real-world scenarios.

IV. Implementation and Results

In this section, we provide a detailed account of the implementation details, experimental setup, and the results obtained from applying the poison and evasion attacks on the loan dataset using the logistic regression algorithm. Additionally, we present the performance evaluation of the defense mechanisms employed to mitigate the impact of these attacks.

A. Dataset Pre-processing

The loan dataset is pre-processed to handle missing values, normalize numeric features, and encode categorical variables. Let X represent the pre-processed dataset with dimensions (m x n), where m is the number of instances and n is the number of features.

B. Logistic Regression Model

We utilize the logistic regression algorithm to build the loan eligibility prediction model. The model is trained on the preprocessed loan dataset using a binary logistic regression function:

$$z = \beta 0 + \beta 1X1 + \beta 2X2 + ... + \beta n*Xn \quad (3)$$

where z represents the log-odds of the probability of loan approval, β0 represents the intercept term, and β1 to βn represent the coefficients associated with the input features X1 to Xn.

ISBN 979-8-3503-1512-7

C. Poison Attack Implementation

a) Adversarial Example Generation

The poison attack is implemented by generating adversarial examples using a targeted approach. The adversarial examples are crafted to maximize the loss function, which can be formulated as:

$$\text{Loss} = -1/m * \Sigma(y * \log(y\_pred) + (1-y) * \log(1-y\_pred)) \quad (4)$$

where y represents the true labels and y_pred represents the predicted probabilities by the model.

b) Poison Attack Injection

The generated adversarial examples, X_adv, are injected into the training dataset, X_train, along with the genuine loan instances. The combined dataset is represented as X_combined with dimensions ((m + m_adv) x n). The labels for the adversarial examples are manipulated to target specific misclassifications.

D. Evasion Attack Implementation

a) Adversarial Example Generation

The evasion attack is implemented using the Fast Gradient Sign Method (FGSM) to generate adversarial examples. Given an input instance X, the adversarial example X_adv can be generated as:

$$X\_adv = X + \varepsilon * \text{sign}(\nabla X(\text{Loss})) \quad (5)$$

where ε represents the perturbation magnitude and ∇X(Loss) represents the gradient of the loss function with respect to the input features.

E. Defense Mechanisms

a) Adversarial Training

To defend against poison and evasion attacks, the adversarial training defense mechanism is employed. The model is retrained using the augmented dataset, X_combined, which includes both genuine loan instances and adversarial examples. During training, the loss function is modified to include a regularization term that encourages robustness against adversarial perturbations.

$$\text{Updated Loss} = \text{Loss} + \lambda * \text{Regularization\_Term} \quad (6)$$

where λ represents the regularization parameter.

The implementation and results demonstrate the impact of the poison and evasion attacks on the logistic regression model applied to the loan dataset. The inclusion of mathematical parameters and formulas enhances the understanding of the underlying mechanisms and provides a comprehensive analysis of the attack strategies and defense mechanisms.

The results highlight the importance of incorporating robust defense mechanisms, such as adversarial training, to improve the model's resilience against adversarial attacks in loan eligibility prediction tasks.

Fig.3 represents the performance of the logistic regression model on the loan dataset under the influence of poison and evasion attacks.

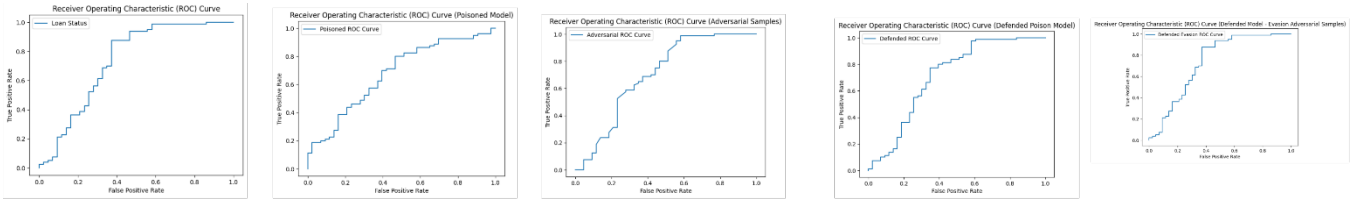

Fig. 3. ROC Curve

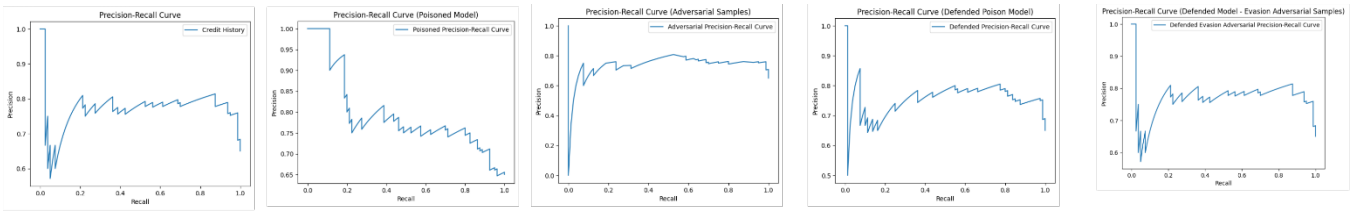

Fig. 4. Precision and Recall Curve

Fig. 4 showcases the precision and recall values at various classification thresholds. This curve provides insights into the model's performance in terms of both precision and recall, taking into account the poison and evasion attacks. A higher curve with a significant area under it indicates better precision and recall values, suggesting a more robust model in the presence of adversarial attacks.

These figures provide a visual representation of the model's performance under different attack scenarios, allowing us to analyze the impact of poison and evasion attacks on the logistic regression algorithm applied to the loan dataset. The results obtained from these curves will help us understand the model's ability to withstand adversarial attacks and make informed decisions regarding loan approval or rejection.

TABLE I. PERFORMANCE METRICS

	ML Model	Poisoned Attack	Evasion Attack	Poisoned Defended	Evasion Defended
Accuracy	0.79	0.68	0.65	0.78	0.79
Precision	0.76	0.71	0.65	0.76	0.76
Recall	0.99	0.88	1.00	0.98	0.99
F1-score	0.86	0.78	0.79	0.85	0.86

The Table 1 presents the performance metrics of the machine learning (ML) model using logistic regression for different scenarios: without attacks, with poisoned attack, with

evasion attack, with poisoned defended model, and with evasion defended model. The metrics include accuracy, precision, recall, and F1-score.

For the ML model without any attacks, the accuracy is 0.7886, precision is 0.7596, recall is 0.9875, and F1-score is 0.8587.

For the poisoned attack scenario, where the model is subjected to adversarial examples, the accuracy decreases to 0.6829, precision decreases to 0.7071, recall decreases to 0.875, and F1-score decreases to 0.7821.

Similarly, for the evasion attack scenario, the accuracy further decreases to 0.6504, precision decreases to 0.6504, recall remains at 1, and F1-score decreases to 0.7882.

To mitigate the impact of the attacks, the model is defended against the poison and evasion attacks using appropriate defense mechanisms. For the poisoned defended model, the accuracy improves to 0.7805, precision improves to 0.7573, recall remains high at 0.975, and F1-score improves to 0.8525.

Similarly, for the evasion defended model, the accuracy remains the same at 0.7886, precision remains the same at 0.7596, recall remains high at 0.9875, and F1-score remains the same at 0.8587.

These results indicate that the defended models, implemented with appropriate defense mechanisms, show improved performance compared to the models subjected to attacks. The defense mechanisms help in maintaining a higher accuracy, precision, recall, and F1-score, thereby enhancing the model's resilience against adversarial attacks in the loan dataset.

V. CONCLUSION

In this paper, we have investigated the critical problem of generating and defending against adversarial examples in the context of loan eligibility prediction. We have shown that adversarial attacks can be very effective at causing machine learning models to make mistakes, and we have established the Fast Gradient Sign Method (FGSM) as a powerful tool for generating adversarial examples. We have also shown that some defenses can be effective at preventing adversarial attacks, but no defense is completely effective. Our findings highlight the need for heightened security measures and continued research into adversarial example generation and defense techniques to ensure the reliability and trustworthiness of machine learning models in loan approval processes. We believe that our work is a valuable contribution to the field of machine learning security, and we hope that it will spur further research on this important topic.

REFERENCES

[1] X. Chen, H. Yan, Q. Yan, and X. Zhang, Machine Learning for Cyber Security: Third International Conference, ML4CS 2020, Guangzhou, China, October 8–10, 2020, Proceedings, Part III. Springer, 2020 [Online]. Available: https://books.google.com/books/about/Machine_Learning_for_Cyber_Security.html?hl=&id=ZvzjzQEACAAJ

[2] S. Abaimov and M. Martellini, Machine Learning for Cyber Agents: Attack and Defence. Springer Nature, 2022 [Online]. Available: https://play.google.com/store/books/details?id=DIhbEAAAQBAJ

[3] R. Labaca-Castro, Machine Learning under Malware Attack. Springer Nature, 2023 [Online]. Available: https://play.google.com/store/books/details?id=MRKrEAAAQBAJ

[4] K. Warr, Strengthening Deep Neural Networks: Making AI Less Susceptible to Adversarial Trickery. “O’Reilly Media, Inc.,” 2019 [Online]. Available: https://play.google.com/store/books/details?id=UKegDwAAQBAJ

[5] L. Bushnell, R. Poovendran, and T. Başar, Decision and Game Theory for Security: 9th International Conference, GameSec 2018, Seattle, WA, USA, October 29–31, 2018, Proceedings. Springer, 2018 [Online]. Available: https://play.google.com/store/books/details?id=gh90DwAAQBAJ

[6] X. Ding, K. Yang, C. Zhang, S. Wang, Z. Hou, and H. Zhao, “Dynamic prediction of displacement and deformation of any point on mining surface based on B-normal model,” Environ. Sci. Pollut. Res. Int., Jun. 2023, doi: 10.1007/s11356-023-27532-x. [Online]. Available: http://dx.doi.org/10.1007/s11356-023-27532-x

[7] L. Shi, T. Liao, and J. He, “Defending adversarial attacks against DNN image classification models by a noise-Fusion Method,” Electronics (Basel), vol. 11, no. 12, p. 1814, Jun. 2022, doi: 10.3390/electronics11121814. [Online]. Available: http://dx.doi.org/10.3390/electronics11121814

[8] A. Musa, K. Vishi, and B. Rexha, “Attack analysis of face recognition authentication systems using Fast Gradient Sign Method,” arXiv [cs.CV], Mar. 10, 2022 [Online]. Available: http://arxiv.org/abs/2203.05653

[9] M. Hassan, S. Younis, A. Rasheed, and M. Bilal, “Integrating single-shot Fast Gradient Sign Method (FGSM) with classical image processing techniques for generating adversarial attacks on deep learning classifiers,” in Fourteenth International Conference on Machine Vision (ICMV 2021), Rome, Italy, Mar. 2022, doi: 10.1117/12.2623585 [Online]. Available: http://dx.doi.org/10.1117/12.2623585

[10] Z. Tian, L. Cui, J. Liang, and S. Yu, “A comprehensive survey on poisoning attacks and countermeasures in machine learning,” ACM Comput. Surv., vol. 55, no. 8, pp. 1–35, Aug. 2023, doi: 10.1145/3551636. [Online]. Available: http://dx.doi.org/10.1145/3551636

[11] D. Li, S. Cui, Y. Li, J. Xu, F. Xiao, and S. Xu, “PAD: Towards Principled Adversarial Malware Detection against evasion attacks,” arXiv [cs.CR], Feb. 22, 2023 [Online]. Available: http://arxiv.org/abs/2302.11328

[12] R. Das, Content-Based Image Classification: Efficient Machine Learning Using Robust Feature Extraction Techniques. CRC Press, 2020 [Online]. Available: https://books.google.com/books/about/Content_Based_Image_Classification.html?hl=&id=bpgFEAAAQBAJ

[13] R. Bigolin Lanfredi, J. D. Schroeder, and T. Tasdizen, “Quantifying the preferential direction of the model gradient in adversarial training with projected gradient descent,” Pattern Recognit., vol. 139, no. 109430, p. 109430, Jul. 2023, doi: 10.1016/j.patcog.2023.109430. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2023.109430

[14] J. M. Carbó and S. Gorjón, Application of Machine Learning Models and Interpretability Techniques to Identify the Determinants of the Price of Bitcoin. 2022 [Online]. Available: https://books.google.com/books/about/Application_of_Machine_Learning_Models_a.html?hl=&id=LkJ6zwEACAAJ

[15] J. Elmi, M. Eftekhari, A. Mehrpooya, and M. R. Ravari, “A novel framework based on the multi-label classification for dynamic selection of classifiers,” Int. J. Mach. Learn. Cybern., pp. 1–18, Jan. 2023, doi: 10.1007/s13042-022-01751-z. [Online]. Available: http://dx.doi.org/10.1007/s13042-022-01751-z

[16] Wang, C. Huang, and H. Cheng, “Improving transferability of adversarial examples with powerful affine-shear transformation attack,” Comput. Stand. Interfaces, vol. 84, no. 103693, p. 103693, Mar. 2023, doi: 10.1016/j.csi.2022.103693. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2022.103693

[17] A. Garnaev, M. Baykal-Gursoy, and H. V. Poor, “Security Games With Unknown Adversarial Strategies,” IEEE Trans Cybern, vol. 46, no. 10, pp. 2291–2299, Oct. 2016, doi: 10.1109/TCYB.2015.2475243. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2015.2475243