MISLEAD: Manipulating Importance of Selected features for Learning Epsilon in Evasion Attack Deception

A Preprint

Vidit Khazanchi
Department of Materials
Indian Institute Technology
Bombay, India
viditk0812@gmail.com

Pavan Kulkarni
AIShield
Bosch Global Software Technologies
Bangalore, India
pavan.kulkarni@in.bosch.com

Yuvaraj Govindarajulu
AIShield
Bosch Global Software Technologies
Bangalore, India
govindarajulu.yuvaraj@de.bosch.com

Manojkumar Parmar
AIShield
Bosch Global Software Technologies
Bangalore, India
manojkumar.parmar@in.bosch.com

May 3, 2024

Abstract

Emerging vulnerabilities in machine learning (ML) models due to adversarial attacks raise concerns about their reliability. Specifically, evasion attacks manipulate models by introducing precise perturbations to input data, causing erroneous predictions. To address this, we propose a methodology combining SHapley Additive exPlanations (SHAP) for feature importance analysis with an innovative Optimal Epsilon technique for conducting evasion attacks. Our approach begins with SHAP-based analysis to understand model vulnerabilities, crucial for devising targeted evasion strategies. The Optimal Epsilon technique, employing a Binary Search algorithm, efficiently determines the minimum epsilon needed for successful evasion. Evaluation across diverse machine learning architectures demonstrates the technique's precision in generating adversarial samples, underscoring its efficacy in manipulating model outcomes. This study emphasizes the critical importance of continuous assessment and monitoring to identify and mitigate potential security risks in machine learning systems.

Keywords Adversarial Machine Learning, Evasion Attacks, SHAP (SHapley Additive exPlanations), Model Vulnerability Assessment, Tabular Data Analysis.

1 Introduction

The widespread adoption of machine learning models has driven remarkable technological advancements and improvements in decision-making, concurrently exposing a vulnerability—adversarial attacks, particularly evasion attacks. These attacks involve subtle alterations to input data, leading to erroneous predictions with potentially severe consequences Szegedy et al. (2014); Goodfellow et al. (2015); Carlini and Wagner (2017). To address this challenge, our approach introduces a methodology combining SHapley Additive exPlanations (SHAP) for feature importance analysis with an innovative optimal epsilon technique Lundberg and Lee (2017). Motivated by the growing need to secure machine learning models in vital sectors like healthcare, finance, autonomous vehicles, and cybersecurity Biggio and Roli (2018), our methodology integrates feature importance analysis using SHAP, a powerful tool for understanding feature impact across various domains Lundberg et al. (2019). This analysis spans both binary and multiclass classification scenarios, offering insights for developing targeted evasion strategies by evaluating the significance of different features.

The optimal epsilon technique, introduced in our study, plays a pivotal role in evasion attacks by determining the minimal epsilon necessary for successful evasion. This concept involves finding the smallest perturbation magnitude to deceive a machine learning model into making incorrect predictions without detectable alterations. The technique enhances precision and effectiveness in exposing vulnerabilities, underscoring the need for robust countermeasures Papernot et al. (2017). Through experiments, we assess the methodology's effectiveness across diverse machine learning architectures and datasets, showcasing its ability to generate precise adversarial samples Wang and He (2021).

The contributions of this paper are as follows:

- **Integration of SHAP with Evasion Attacks:** The novel approach of systematically integrating SHAP-based feature importance analysis into the evasion attack process, allowing for targeted manipulation of the most influential features, leading to more efficient and effective attacks Huang et al. (2017); Dvijotham et al. (2018).
- **Optimal Epsilon Technique:** Introduction of a novel and systematic technique for determining the minimum epsilon needed for successful evasion through a binary search-based approach, enhancing the precision of adversarial sample generation and providing a nuanced understanding of model robustness Athalye et al. (2018).
- **Black-Box Applicability:** MISLEAD operates in a black-box setting, relying solely on the model's predictions, making it applicable to real-world scenarios where attackers might not have access to the model's internal parameters.
- **Comprehensive Feature Analysis:** Thorough analysis of feature impacts, categorizing them based on their influence and directionality, allowing for the development of sophisticated and targeted attack strategies.

The paper is organized as follows: Section 2 explores fundamental theories and previous studies, providing a base for our research approach. In Section 3, we detail our methodology, including SHAP-based feature importance analysis and the innovative optimal epsilon technique for evasion attacks. Section 4 discusses our experimental setup and findings, highlighting the effectiveness and implications of our work. Finally, Section 5 concludes the paper with a summary of our findings and potential avenues for future research.

2 Background

2.1 Adversarial Machine Learning

Recent years have underscored machine learning models' vulnerability to adversarial attacks, especially evasion attacks that involve crafting adversarial examples for specific target class predictions. Adversarial attacks categorize based on the attacker's knowledge: perfect (white-box), limited (gray-box), and zero knowledge (black-box) attacks Nazemi and Fieguth (2019); Hitaj et al. (2017); Sotgiu et al. (2020); Biggio and Roli (2018). These categories depend on the attacker's understanding of training data and model parameters.

2.2 Feature Importance

SHAP has emerged as a powerful tool for understanding machine learning models' decision-making process Lundberg and Lee (2017). Applied across domains, including image classification, natural language processing, and tabular data analysis Marcílio and Eler (2020); Panati et al. (2022); Mosca et al. (2022); Lundberg et al. (2020), SHAP values enhance interpretability, aiding feature selection and model optimization Cai et al. (2018); Chen et al. (2018); Ancona et al. (2018). The optimal epsilon technique in our paper systematically determines the minimum epsilon for effective evasion, a valuable contribution to the field. While epsilon's role in controlling perturbation magnitude is well-discussed, systematic techniques for determining optimal epsilon are underexplored. Our paper adapts binary search algorithms, a novel approach in evasion attacks Yu et al. (2017); Han and Lu (2012/09); Meyers et al. (2023).

2.3 Evasion attacks on ML models

Fast Gradient Sign Method (FGSM) is one of the pioneering techniques in evasion attacks. It computes gradients with respect to the input data and perturbs the data in the direction that maximizes the loss, thus causing misclassification Goodfellow et al. (2015). Projected Gradient Descent (PGD) is an iterative variant of FGSM that performs multiple steps of gradient descent while ensuring that the perturbed data remains within an epsilon ball around the original sample Madry et al. (2017a). DeepFool is an attack method that computes the perturbation by linearizing the decision boundary of the model and iteratively finding the closest decision boundary point Moosavi-Dezfooli et al. (2016). Papernot et. al. have explored practical black-box attacks against machine learning models, emphasizing the real-world applicability of adversarial attacks Papernot et al. (2017). Athalye et. al. investigated techniques for synthesizing robust adversarial examples, aiming to create adversarial samples that are less susceptible to detection and defense mechanisms Athalye et al. (2018).

2.4 Feature + Evasion on Tabular classification

In tabular datasets, attacks focus on domain-specific challenges, like financial datasets Hashemi and Fathi (2020); Sarkar et al. (2018); Cartella et al. (2021). Novel methods, such as Max Salience Attack (MSA), aim to minimize altered features Sarkar et al. (2018). Our paper proposes a unique methodology integrating SHAP-based feature importance analysis into evasion attacks, providing a comprehensive perspective on model vulnerabilities. This differs from recent advancements like Feature Importance Guided Attack (FIGA) Gressel et al. (2023), emphasizing minimal perturbation using SHAP and an optimal epsilon technique.

2.5 SHAP for Black Box Access

Hassija et al., Hassija et al. (2024) explains, SHAP operates as a black-box explainer for black-box models:

- **SHAP's Functionality:** SHAP focuses on explaining individual predictions, not the entire inner workings of the model.
- **Model Agnostic:** The key strength of SHAP lies in its model-agnostic nature. It doesn't require knowledge of the model's architecture (e.g., decision trees, neural networks) to compute feature contributions. It treats the model as a function, taking inputs and generating outputs.
- **SHAP's Internal Workings:** SHAP leverages game theory concepts (Shapley Values) to fairly distribute credit for a prediction amongst all features. While the underlying calculations involve the model's predictions for various data permutations, SHAP itself remains agnostic to the specific model logic.

In essence, SHAP acts as an intermediary. It interacts with the black-box model at the input-output level, extracting feature importance without needing to delve into the internal complexities of the model. This allows SHAP to provide valuable insights into a model's decision-making process without requiring white-box access.

3 Methodology

This section provides a detailed explanation of our evasion attack methodology, covering the overall threat model, key assumptions, data collection, and preprocessing procedures. The goal is to enhance the reliability and quality of our input data, emphasizing factors influencing the model's predictions through SHAP techniques for feature importance analysis. Following this analysis, we categorize the impact of individual features based on our findings and introduce an attack strategy, a carefully designed plan to systematically modify input samples, outsmarting the model's predictions to achieve the desired target class.

3.1 Threat Model and Assumptions

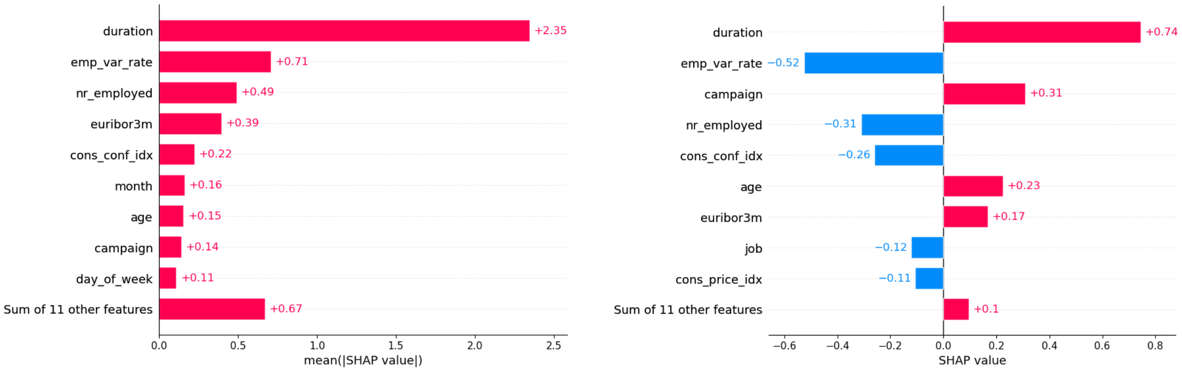
Our research assumes the existence of a machine learning target model vulnerable to evasion attacks due to its sensitivity to changes in input features. Focusing on tabular datasets with numerical and categorical data, our attack strategy operates in a complete black-box setting, where the attacker can only query the target model for predictions without insight into internal parameters and weights.

3.2 Data Collection and Preprocessing

We employ a Bank Marketing dataset comprising numerical and categorical variables. The class distribution shows 36,548 samples in Class 0 and 4,640 samples in Class 1. With 10 numerical and 10 categorical features, the dataset is comprehensively characterized. Categorical variables are numerically encoded using LabelEncoder Pedregosa et al. (2011), and both numerical and categorical features are normalized to a 0 to 1 range using MinMaxScaler Pedregosa et al. (2011) for comparative analysis.

3.3 Feature Importance Analysis using SHAP

SHAP values quantify the influence of each feature on model predictions, providing insights into the direction and magnitude of their impact. Calculating SHAP values for every sample generates an array of values for the 20 features.



(a) Global Bar Plot

(b) Local Bar Plot

Figure 1: Binary Classification Bar Plots

These insights are analyzed through various plots to rank features, assess their average impact, and identify the most influential ones in the dataset. The application of SHAP is explored in both binary and multiclass classification scenarios.

3.3.1 Binary Classification

In binary classification, involving only two labels, interpreting SHAP values is more straightforward. The following plots are employed:

Global Bar Plot: Offers a comprehensive view of feature importance across the entire dataset. It displays the mean SHAP value for each feature, arranged in descending order of importance, highlighting influential features in driving model predictions (Figure 1(a)). However, it doesn't indicate the direction of impact or specific feature values necessary for predicting specific classes.

Local Bar Plot: Zooms in on individual samples, detailing how each feature impacts the model's prediction for a specific sample. This plot enhances understanding at the micro-level, revealing intricate relationships between features and predictions (Figure 1(b)).

Beeswarm Plot: Figure 2 provides a more detailed and informative visualization than the bar plots. It showcases the relative importance of features and their relationship with the predicted outcome, offering a comprehensive overview of how variables influence predictions. This insight is critical for generating perturbations in our evasion attack strategy.

Together, these plots form an integral part of our methodology, enabling analysis of the impact of individual features on model predictions in both macro and micro perspectives.

3.3.2 Multiclass Classification

Multiclass classification introduces more complexity with multiple classes for prediction. The following plots are applied:

Global Bar Plot: Represents each feature with a bar divided into sections corresponding to each class (Figure 3(a)). This allows for a detailed understanding of a feature's importance across different classes, revealing the variable significance of features in predicting various classes.

Beeswarm Plot: Similar to the binary classification scenario, the multiclass beeswarm plot provides an intricate visualization of SHAP values across the dataset. However, a notable difference is the existence of separate graphs for each class (Figures 3(b), 3(c), 3(d)), illustrating how features influence predictions for a specific class. This detailed representation is key to understanding the optimal feature values necessary for predicting each class, crucial for devising targeted evasion strategies.

Together, these plots in multiclass classification scenarios enable a comprehensive analysis of feature impacts, both across and within individual classes. They enhance understanding of how different features contribute to model predictions in multiclass settings, vital for developing sophisticated evasion techniques in a multiclass context.

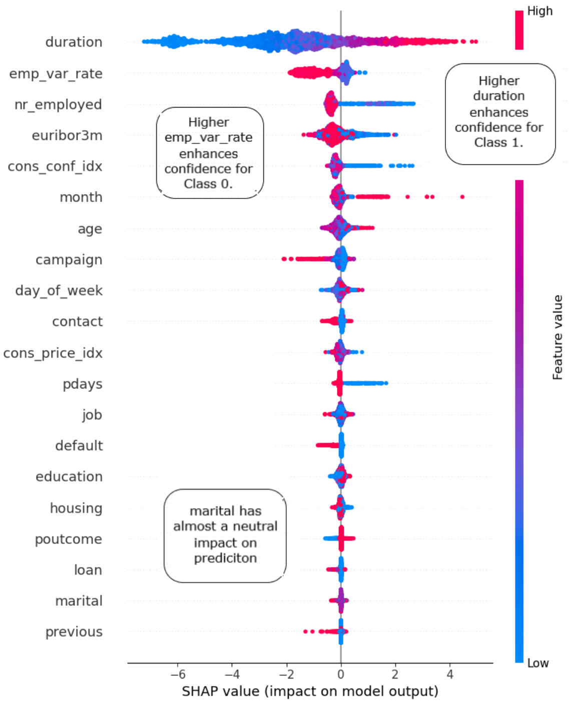
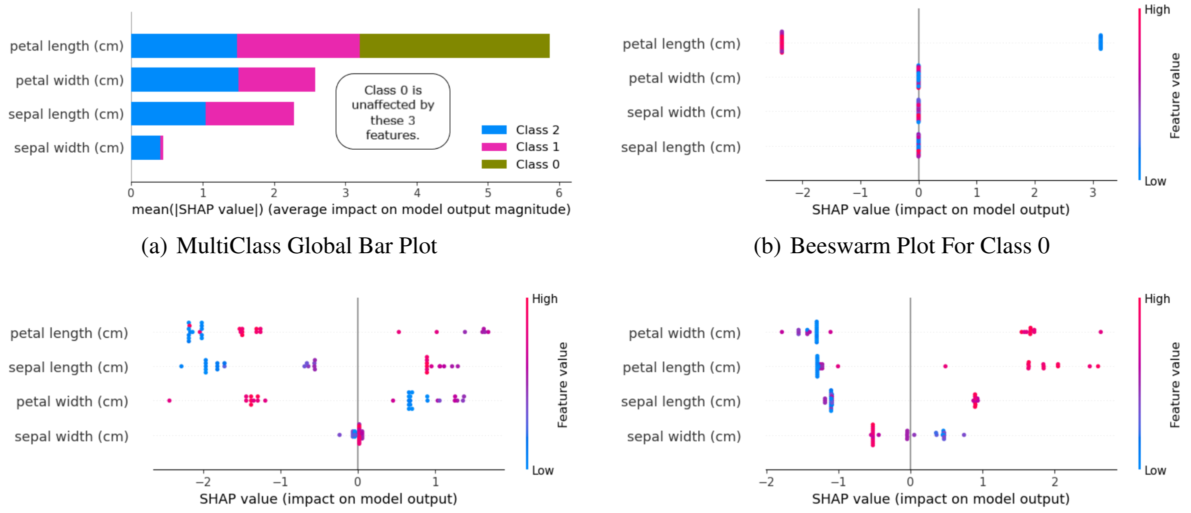


Figure 2: Binary Classification Beeswarm Plot



(a) MultiClass Global Bar Plot

(b) Beeswarm Plot For Class 0

(c) Beeswarm Plot For Class 1

(d) Beeswarm Plot For Class 2

Figure 3: MutliClass Classification Beeswarm and Global Bar Plots

3.4 Feature Analysis for Evasion

In addressing the challenge of evasion attacks, we conduct a comprehensive analysis of feature impacts on model predictions. This section details our approach to categorize and utilize feature behavior for devising evasion strategies. By understanding how individual features influence model predictions, we aim to identify vulnerabilities in machine learning models and exploit these for successful evasion.

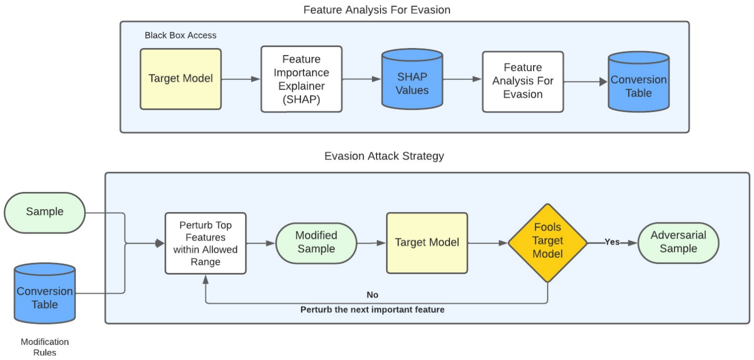


Figure 4: Feature Analysis For Evasion

3.4.1 Feature Impact Categorization

Our methodology begins with categorizing the impacts of individual features, pivotal for understanding their influence on model predictions. We use predefined thresholds, denoted as T_{low} and T_{high}, to assign impact categories: 'Low' (L), 'Medium' (M), or 'High' (H), based on the feature's value F_{ij} using Equation 1.

$$CF_{ij} = \begin{cases} L & \text{if } F_{ij} < T_{low} \\ M & \text{if } T_{low} \leq F_{ij} < T_{high} \\ H & \text{if } F_{ij} \geq T_{high} \end{cases} \tag{1}$$

where, CF_{ij} represents the impact category of feature i in sample j

3.4.2 Categorizing SHAP Values

Additionally, we categorize the SHAP values S_{ij}, labeling them as 'positive' (P), 'neutral' (N_T), or 'negative' (N), based on their sign using Equation 2.

$$CS_{ij} = \begin{cases} P & \text{if } S_{ij} > 0 \\ N_T & \text{if } S_{ij} = 0 \\ N & \text{if } S_{ij} < 0 \end{cases} \tag{2}$$

where, CS_{ij} represents the categorized SHAP value for feature i in sample j

3.4.3 SHAP Summary Dictionary

Next, we initiate a SHAP summary dictionary (SSD) to capture the impact of features i for each class c in the dataset using Equation 3.

$$SSD = \{c : \{i : \{CS : []\}, ...\}, ...\} \tag{3}$$

where, CS represents $\{P, N_T, N\}$

Each class c, feature i, and SHAP category (CS) is represented, with the impact category (CF) accumulated in a list.

3.4.4 Concise SHAP Summary Direction

Using a count function, we quantify occurrences of impacts for each feature within each class. The sentiment (CS) with the maximum count for each impact (CF) and feature (i) in class (c) is then determined using Equations 4 and 5.

$$M_{c,i}(CF) = \arg\max_{CS} C_{i,j,CS}(CF) \tag{4}$$

$$C_{c,i,CS}(CF) = \text{\# of times } CF \text{ appears in } CS \tag{5}$$

This aggregation process leads to a more concise SSD, enabling us to better understand the relationship between features and their impacts across different classes. Appendix A Figure 8(a) shows the Concise SSD for the Iris Dataset.

3.4.5 Possible Class Conversions

The set of possible class conversions, denoted as $class_{conversions}$, is a set containing pairs (i, j) where i and j are unique classes, and $i \neq j$. It includes all possible combinations of unique class pairs, ensuring that each pair consists of different classes, as defined by Equation 6.

$$class_{conversions} = \{(i, j)|i, j \in \text{classes and } i \neq j\} \tag{6}$$

3.4.6 Feature Impact Aggregation

To determine how feature impacts from one class i can be converted to another j, an intersection method is used. This method involves intersecting the $negative$ and $neutral$ impacts for class i (I_i^-) with the $positive$ impacts for class j $(positive_j)$. This intersection results in P_{ij} which indicates a strong positive effect on class j while exerting a negative impact on class i.

Positive Effect on Class j:

$$P_{ij} = I_i^- \cap \text{positive}_j \tag{7}$$

Conversely, the $positive$ and $neutral$ impacts for class j (I_j^+) are intersected with the $negative$ impacts for class i $(negative_i)$. This intersection yields N_{ij}, signifying a strong negative effect on class i with a positive effect on class j.

Negative Effect on Class i:

$$N_{ij} = I_j^+ \cap \text{negative}_i \tag{8}$$

The final step involves taking the union of P_{ij} and N_{ij}. This union synthesizes the modifications necessary to promote a positive effect on class j while simultaneously inducing a negative effect on class i.

Final Effect to move from Class i to Class j:

$$F_{ij} = P_{ij} \cup N_{ij} \tag{9}$$

Such modifications are critical in performing the targeted evasion attack, effectively moving from class i to class j.

3.4.7 Storing Conversion Directions

In the final step of our process, we store conversion directions for each feature in the conversion table. These directions are pivotal in guiding modifications to feature values during a targeted evasion attack. Appendix A Figure 8(b) shows the conversion table for the Iris Dataset.

This table plays a pivotal role in mapping the impact of each feature from the original class to its potential impact on a target class, contingent upon modifications in feature values. The entire process of feature analysis for Evasion is encapsulated in the 'Feature Analysis Block' of Figure 4, providing a visual representation of the methodology and its components.

3.5 Evasion Attack Strategy with Optimal Epsilon Technique

Our evasion attack strategy (refer Appendix B, Algorithm 1) operates in a black-box setting, relying solely on the model's output predictions to guide the iterative process of modifying input features. The primary objective is to manipulate a given input sample, x_{org}, such that it deceives the machine learning model into misclassifying it as a target class, different from its original class.

This strategy leverages the knowledge acquired through the comprehensive feature importance analysis using SHAP, as detailed in Section 3.3 and Section 3.4. The conversion table, derived from this analysis, provides crucial insights into the directional adjustments required for each feature to facilitate the desired class conversion.

The evasion attack algorithm iterates over the features of the input sample, modifying them according to the conversion rules specified in the conversion table. The adjustments, denoted as Δx_i, are carefully calibrated to remain within plausible bounds, ensuring that the modifications do not exceed a predefined threshold, d_{max}, also referred to as the epsilon (ϵ) value:

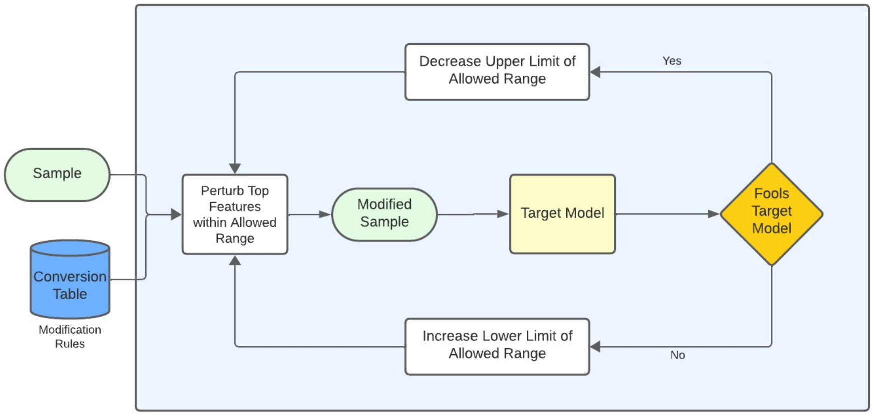

Decrease Upper Limit of Allowed Range
Yes
Sample
Perturb Top Features within Allowed Range
Modified Sample
Target Model
Fools Target Model
Conversion Table
Modification Rules
Increase Lower Limit of Allowed Range
No

Figure 5: Optimal Epsilon

$$x'_i = x_i + \Delta x_i \tag{10}$$

During the attack process, we continuously monitor the distance between the modified sample, $x_{\text{adv\_temp}}$, and the original input, x_{org}, represented as distance$(x_{\text{adv_temp}}, x_{\text{org}})$. The objective is to find the adversarial sample, $x_{\text{adv\_best}}$, that successfully triggers misclassification into the target class with minimal deviation from the original sample

$$x_{\text{adv\_best}} = \arg\min_{x_{\text{adv\_temp}}} \text{distance}(x_{\text{adv\_temp}}, x_{\text{org}}) \tag{11}$$

To refine the evasion attack approach and ensure the generation of effective adversarial samples with minimal perturbation, we introduce the Optimal Epsilon technique, as shown in Figure 5. This technique systematically determines the smallest epsilon ($\epsilon_{\text{optimal}}$) necessary for successful evasion by employing a binary search loop, refer to Algorithm 2 in Appendix B.

The binary search process starts with an initial epsilon range $[\epsilon_{\text{low}}, \epsilon_{\text{high}}] = [0, 0.5]$, where 0.5 represents the upper limit for allowed perturbation. This upper limit can be extended to 1 for determining the optimal epsilon across all samples without restrictions on perturbation magnitude. The loop iterates until the gap between ϵ_{high} and ϵ_{low} is less than a predefined $tolerance$ value.

Within each iteration, adversarial samples are generated by modifying feature values according to the conversion rules, aiming to shift the prediction from the original class to the target class. The effectiveness of these samples is evaluated on the target model. If a successful adversarial sample is found, the distance between the adversarial and original samples is calculated, and the optimal epsilon and the best adversarial sample are updated accordingly.

The binary search concludes once the difference between ϵ_{high} and ϵ_{low} falls below the $tolerance$ threshold. The final $\epsilon_{\text{optimal}}$ represents the minimum perturbation magnitude required for effective adversarial samples under the given conditions.

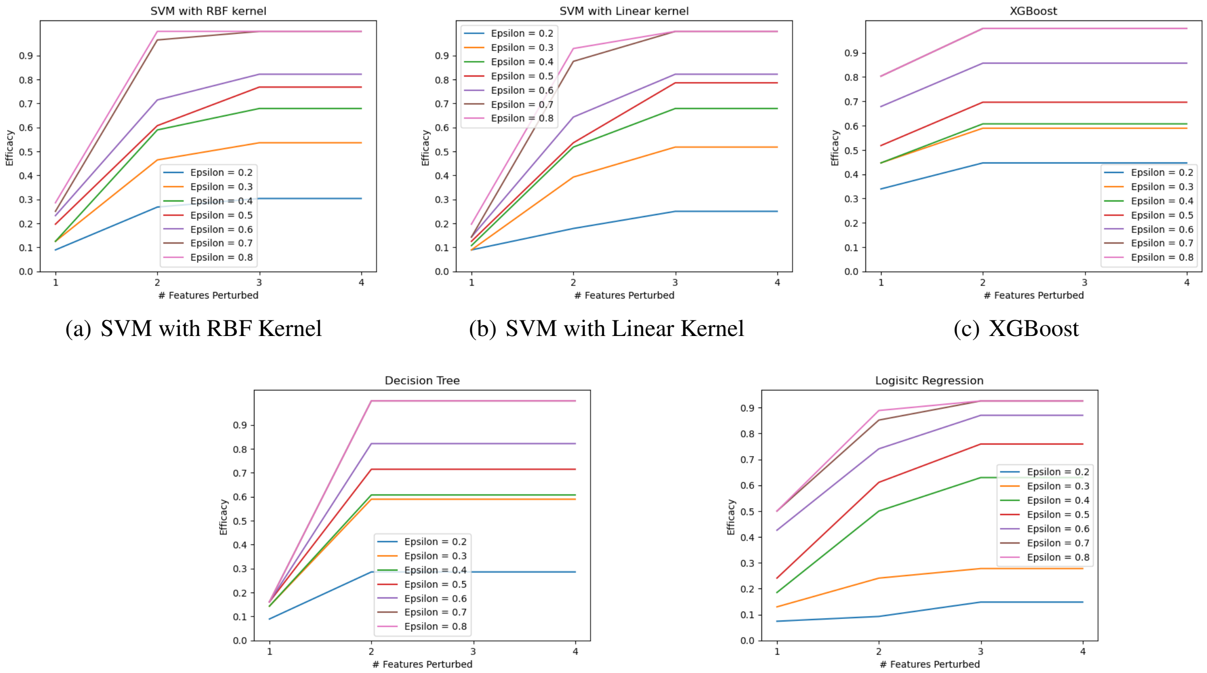
By integrating the Optimal Epsilon technique seamlessly into the evasion attack strategy, our methodology ensures the generation of precise and impactful adversarial samples, underscoring the vulnerability of machine learning models to carefully crafted evasion attacks.

4 Experiments

4.1 Efficacy as a Key Metric for Evasion

Unlike traditional accuracy metrics in machine learning, efficacy in the context of our paper refers to the model's susceptibility to evasion attacks. Specifically, it measures the proportion of samples that successfully evade and deceive the model within the allowed perturbation limit, denoted as epsilon.

Let N be the total number of samples subjected to the evasion attack. Out of these, let N_{evaded} be the number of samples that successfully evade the model's detection within the perturbation limit. The efficacy, E, can then be mathematically represented as:



(a) SVM with RBF Kernel

(b) SVM with Linear Kernel

(c) XGBoost

(d) Decision Tree

(e) Logistic Regression

Figure 6: Saturation Point: MultiClass Attacks on Iris Dataset

Table 1: Efficacy of MultiClass Targeted Attack on Iris Dataset

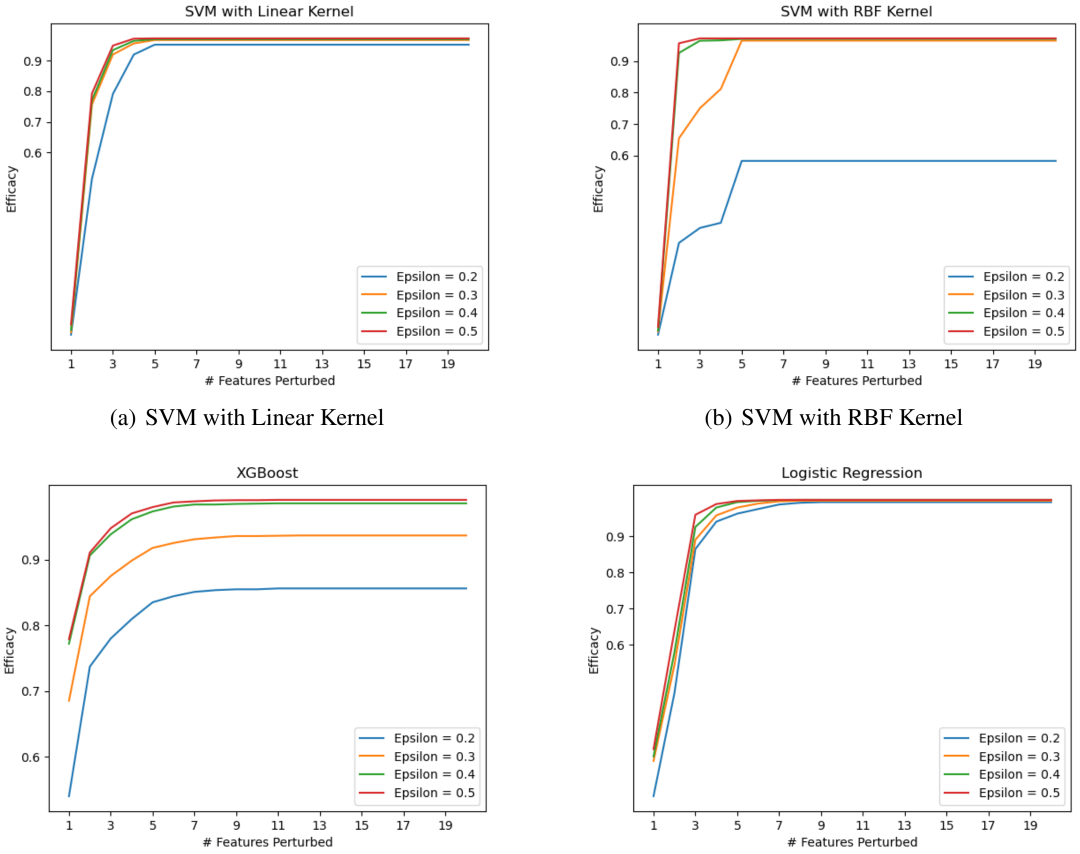
Epsilon	SVM (RBF)			SVM (Linear)			XGBoost			Decision Tree			Logistic Regression		
(ϵ)	Class			Class			Class			Class			Class		
	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
0.3	0.31	0.86	0.33	0.31	0.86	0.28	0.31	1	0.29	0.31	0.96	0.29	0.4	0.33	0.11
0.4	0.62	1	0.4	0.625	1	0.33	0.37	1	0.29	0.37	1	0.29	0.73	0.95	0.16
0.5	0.94	1	0.4	1	1	0.33	0.69	1	0.29	0.69	1	0.29	1	1	0.28
0.6	1	1	0.44	1	1	0.44	0.93	1	0.59	0.94	1	0.59	1	1	0.61

$$E = \frac{N_{evaded}}{N} \tag{12}$$

A high efficacy score implies a greater number of samples evading the model successfully, indicating a potential vulnerability in the model's defense against adversarial attacks within the specified perturbation limit. By evaluating efficacy across different models and configurations, we can compare their robustness against evasion attacks, offering insights into the effectiveness of various defense mechanisms.

4.2 Multiclass Classification - Iris Dataset

In the context of multiclass classification, we extended our research to the Iris dataset Fisher (1988), known for its suitability in multiclass tasks. We evaluated five machine learning architectures: SVM with RBF and Linear kernels, XGBoost, Logistic Regression, and Decision Tree Unwin and Kleinman (2021). Our focus was on perturbing features of original samples to generate adversarial samples, aiming at specific class misclassifications. In the multiclass setting, this approach introduced additional complexity due to the presence of multiple target classes. Table 1 illustrates the efficacy of targeted attacks across different machine learning models and varying epsilon values. Figure 6 visualizes the relationship between the efficacy of attacks and the number of features perturbed. This provides deeper insights into



(a) SVM with Linear Kernel

(b) SVM with RBF Kernel

(c) XGBoost

(d) Logistic Regression

Figure 7: Saturation Point : Binary Class Attack on Bank Marketing Dataset

Table 2: Efficacy of Binary Class Attack on Bank Marketing Dataset

Epsilon (ϵ)	SVM (RBF)	SVM (Linear)	XGBoost	Logistic Regression
0.2	0.58	0.95	0.86	0.99
0.3	0.96	0.96	0.94	1
0.4	0.97	0.97	0.98	1
0.5	0.97	0.97	0.99	1

the optimization of feature perturbations for successful evasion, showing how certain models reach a saturation point beyond which additional perturbations do not significantly increase the success of the attack.

4.3 Binary Classification - Bank Marketing Dataset

In the binary classification context, experiments were carried out on a bank marketing dataset Moro et al. (2012), a subset derived from the original Bank Marketing dataset from the UCI repository Sérgio Moro and Rita (2014). These experiments yielded crucial insights into the robustness of various machine learning models and the dynamics of feature perturbations. We evaluated four machine learning architectures: SVM with Radial Basis Function (RBF) and Linear kernels Boser et al. (1992), XGBoost Chen and Guestrin (2016), Logistic Regression.

Table 2 details the diverse performance spectrum across various models with increasing epsilon values. Notably, Logistic Regression exhibited high evasion susceptibility, reaching perfect evasion (efficacy of 1) at an epsilon value of 0.3 and maintaining this across higher epsilon values. In contrast, SVM with RBF kernel and XGBoost models demonstrated a

Table 3: Comparative Analysis of the Efficacy of Targeted Attacks on the SVM Model for Iris Dataset

Epsilon	Class 0			Class 1			Class 2		
(ϵ)	FGM	PGD	MISLEAD	FGM	PGD	MISLEAD	FGM	PGD	MISLEAD
0.2	0.02	**0.04**	0.03	**0.58**	0.22	0.35	0.28	0.25	**0.4**
0.3	0.08	0.06	**0.31**	0.6	0.33	**0.86**	0.28	0.31	**0.4**
0.4	0.2	0.1	**0.62**	0.7	0.56	**1**	0.28	0.33	**0.4**

Table 4: Comparative Analysis of the Efficacy of Untargeted Attacks on the SVM Model for Iris Dataset

Epsilon (ϵ)	FGM	PGD	MISLEAD
0.2	0.34	0.52	**0.53**
0.3	0.48	0.68	**0.86**
0.4	0.78	0.79	**1**

gradual increase in their susceptibility to evasion attacks as epsilon increased, suggesting a more robust stance against smaller perturbations but a vulnerability at higher epsilon levels. Figure 7 visually illustrates the saturation point in the number of features necessary for successful evasion attacks across different epsilon value. The saturation point is a critical concept, denoting the threshold beyond which increasing the number of perturbed features does not significantly enhance the success rate of the evasion attack.

4.4 Comparative Study

In our comparative study, we evaluate the performance of the MISLEAD technique against established adversarial defense methods, leveraging Fast Gradient Method (FGM) and Projected Gradient Descent (PGD) from Adversarial Robustness Toolbox (ART) Nicolae et al. (2018) and SecML Melis et al. (2019) libraries.

For the Targeted Attack, experiments were conducted with each of the three Iris classes as the target, as shown in Table 3. MISLEAD consistently shows enhanced resilience against targeted attacks towards specific classes when compared to FGM and PGD.

Across various epsilon values in the Untargeted Attack, as shown in Table 4, MISLEAD demonstrates competitive efficacy when compared to FGM and PGD. The qualitative analysis suggests that MISLEAD achieves robust results, especially at higher epsilon values, surpassing existing methods. Importantly, FGM and PGD are considered White Box attacks, while MISLEAD operates as a Black Box attack. This distinction adds a critical layer to the comparative analysis, as MISLEAD's efficacy under limited information about the model internals is a significant aspect in real-world scenarios.

These results collectively suggest that the MISLEAD technique, as a novel approach, displays promising performance in both attack scenarios. The qualitative insights emphasize its potential in providing robust adversarial defense, showcasing its superiority over existing methods across various attack scenarios on tabular data.

4.5 Assessment on Model Accuracy

Upon applying our evasion attack method, as detailed in Tables 5 and 6, we observe a decrease in accuracy models. For the iris dataset, accuracy drops from a stable 0.92 down to 0.00 with progressive increases in the perturbation ϵ. This showcases the attack's capacity to significantly disrupt model performance. The bank marketing dataset shows a similar pattern, with accuracy falling from 0.96 to 0.00. These findings demonstrate our method's ability to highlight the vulnerabilities of machine learning models and stress the need for enhanced defensive measures.

While the current study focuses on tabular data, we believe the MISLEAD methodology can be extended to other data domains, such as images and audio, by leveraging appropriate model explanation techniques. For image data, methods like Grad-CAM Selvaraju et al. (2017), DeepLIFT Shrikumar et al. (2017), and SHAP for images can provide explainable feature representations. Similarly, for audio data, techniques like Layer-wise Relevance Propagation Montavon et al. (2019) can extract interpretable features. By integrating these domain-specific feature importance analysis tools, the MISLEAD approach can identify vulnerabilities and generate targeted adversarial samples across diverse data modalities. This extensibility underscores the generalizability of the proposed methodology, fostering its applicability in securing machine learning systems handling various data types.

Table 5: Impact of Evasion Attacks on Model Accuracy - Iris Dataset

(ϵ)	SVM [RBF]		XGBoost		Logistic Regression	
	Before	After	Before	After	Before	After
0.2	0.92	0.44	0.96	0.48	0.94	0.56
0.3	0.92	0.12	0.96	0.02	0.94	0.36
0.4	0.92	0.00	0.96	0.00	0.94	0.06
0.5	0.92	0.00	0.96	0.00	0.94	0.00

Table 6: Impact of Evasion Attacks on Model Accuracy - Bank Marketing Dataset

	SVM [RBF]		XGBoost		Logistic Regression	
(ϵ)	Before	After	Before	After	Before	After
0.2	0.91	0.38	0.96	0.16	0.91	0.04
0.3	0.91	0.03	0.96	0.08	0.91	0.03
0.4	0.91	0.03	0.96	0.01	0.91	0.00
0.5	0.91	0.03	0.96	0.00	0.91	0.00

5 Mitigation Strategies

Mitigating evasion attacks on AI models is an active area of research, with several promising approaches. Adversarial training, as described by Madry et al. in Madry et al. (2017b), exposes the model to both clean and adversarially crafted data, improving its robustness to slight variations used in evasion attempts. Defensive distillation, proposed by Goldblum et al. in Goldblum et al. (2020), leverages a pre-trained, robust model to train a new model to inherit that robustness. These techniques can be complemented by ensuring high-quality training data with inherent variations and employing ensemble methods for a more robust overall system. Continuous monitoring and adaptation to evolving threats remain essential for maintaining a strong defense.

6 Conclusion

In this work, we have introduced a groundbreaking methodology that combines SHAP-based feature importance analysis with an innovative optimal epsilon technique, significantly amplifying the effectiveness of evasion attacks on machine learning models. This methodology is distinct in its capability to precisely identify and manipulate the most influential features of a learning model, thereby refining the accuracy of adversarial sample generation. Our study's cornerstone, the optimal epsilon technique, determines the minimal perturbation required for successful evasion, optimizing the evasion process and establishing a new benchmark in adversarial attack precision. Employing the SHAP framework, our approach not only deepens the understanding of model vulnerabilities but also facilitates the creation of targeted and highly effective adversarial samples, marking a novel advancement in the field.

Looking towards the future, several research avenues present themselves. One promising direction is the extension of our techniques to different data forms, including image and audio data, to comprehensively assess the vulnerabilities across various machine learning models. Additionally, applying our methods to advanced machine learning architectures, especially deep learning models, could provide invaluable insights into both offensive and defensive strategies in model security. Moreover, the development of robust defense mechanisms against the sophisticated evasion attacks demonstrated in our work stands as a critical area of future exploration. Through these endeavors, we aim to contribute to the ongoing advancement in the security of machine learning systems against increasingly sophisticated adversarial threats.

References

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *2nd International Conference on Learning Representations (ICLR)*, 2014.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations, (ICLR)*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. doi:10.1109/SP.2017.49.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems 30*, pages 4765–4774, 2017.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, December 2018. ISSN 0031-3203.

Scott Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, page 56–67, 05 2019.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, page 506–519, 2017.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1924–1933, June 2021.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv*, 2017.

Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelović, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. Training verified learners with learned verifiers. *ArXiv*, abs/1805.10265, 2018.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *Proceedings of the 35th International Conference on Machine Learning*, 80:284–293, 10–15 Jul 2018.

Amir Nazemi and Paul Fieguth. Potential adversarial samples for white-box attacks. *arXiv preprint arXiv:1912.06409*, 2019.

Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, page 603–618, 2017.

Angelo Sotgiu, Ambra Demontis, Marco Melis, Battista Biggio, Giorgio Fumera, Xiaoyi Feng, and Fabio Roli. Deep neural rejection against adversarial examples. *EURASIP Journal on Information Security*, 2020.

Wilson E. Marcílio and Danilo M. Eler. From explanations to feature selection: assessing shap values as feature selection mechanism. *33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 340–347, 2020. doi:10.1109/SIBGRAPI51738.2020.00053.

Chandana Panati, Simon Wagner, and Stefan Brüggenwirth. Feature relevance evaluation using grad-cam, lime and shap for deep learning sar data classification. *23rd International Radar Symposium (IRS)*, pages 457–462, 2022.

Edoardo Mosca, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh. SHAP-based explanation methods: A review for NLP interpretability. *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4593–4603, October 2022.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018. ISSN 0925-2312.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *In Proceedings of the 37th International Conference on Machine Learning (ICML)*, 80, 2018.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus H. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *International Conference on Learning Representations (ICLR)*, 2018.

Guo Yu, Ruimin Shen, Jinhua Zheng, Miqing Li, Juan Zou, and Yuan Liu. Binary search based boundary elimination selection in many-objective evolutionary optimization. *Applied Soft Computing*, 60:689–705, 2017.

Bo Han and Yongquan Lu. Research on optimization and parallelization of optimal binary search tree using dynamic programming. *Proceedings of the 2nd International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT 2012)*, pages 229–233, 2012/09.

Charles Meyers, Tommy Löfstedt, and Erik Elmroth. Safety-critical computer vision: an empirical survey of adversarial evasion attacks and defenses on computer vision systems. *Artificial Intelligence Review*, pages 1–35, 06 2023.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *6th International Conference on Learning Representations, ICLR Conference Track Proceedings*, 06 2017a.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016. doi:10.1109/CVPR.2016.282.

Masoud Hashemi and Ali Fathi. Permuteattack: Counterfactual explanation of machine learning credit scorecards. *arXiv:2008.10138*, 2020.

Suproteem K. Sarkar, Kojin Oshiba, Daniel Giebisch, and Yaron Singer. Robust classification of financial risk. *arXiv:1811.11079*, 2018.

Francesco Cartella, Orlando Anunciacao, Yuki Funabiki, Daisuke Yamaguchi, Toru Akishita, and Olivier Elshocht. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *arXiv:2101.08030*, 2021.

Gilad Gressel, Niranjan Hegde, Archana Sreekumar, Rishikumar Radhakrishnan, Kalyani Harikumar, Anjali S., and Krishnashree Achuthan. Feature importance guided attack: A model agnostic adversarial attack. *arXiv preprint arXiv:2106.14815*, 2023.

Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting black-box models: A review on explainable artificial intelligence. *Cognitive Computation*, 16(1):45–74, 2024. doi:10.1007/s12559-023-10179-8. URL `https://doi.org/10.1007/s12559-023-10179-8`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

R. A. Fisher. Iris. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.

Antony Unwin and Kim Kleinman. The Iris Data Set: In Search of the Source of Virginica. *Significance*, 18(6):26–29, 11 2021. URL `https://doi.org/10.1111/1740-9713.01589`.

Sérgio Moro, Paulo Cortez, and Paulo Rita. Bank Marketing. UCI Machine Learning Repository, 2012. https://doi.org/10.24432/C5K306.

Paulo Cortez Sérgio Moro and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, page 144–152, 1992.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, 2016.

Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018. URL `https://arxiv.org/pdf/1807.01069`.

Marco Melis, Ambra Demontis, Maura Pintor, Angelo Sotgiu, and Battista Biggio. secml: A python library for secure and explainable machine learning. *arXiv preprint arXiv:1912.10013*, 2019.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi:10.1109/ICCV.2017.74.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org, 2017.

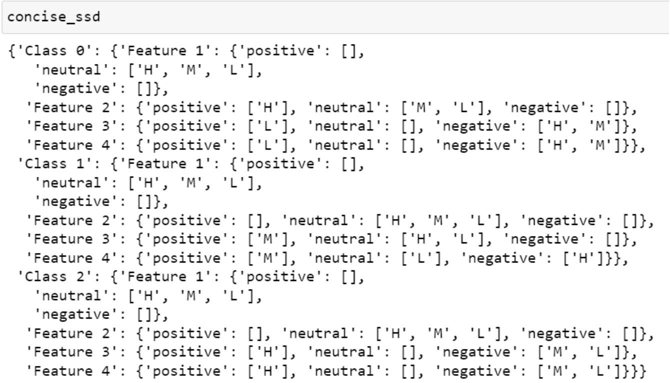
Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017b.

Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3996–4003, 2020.

A Concise SSD and Conversion Table

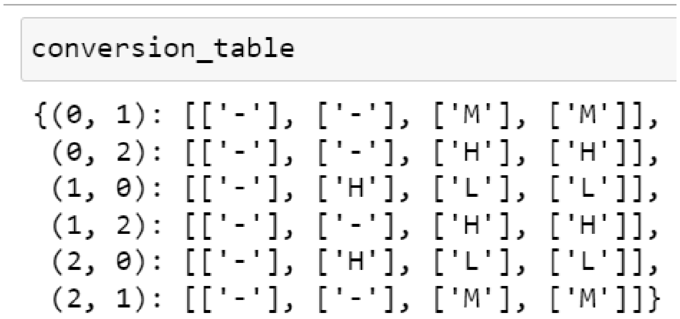
In this section we present the compact SSD and the conversion table for the Iris Dataset. In this representation, the keys denote class conversions, while the corresponding list values illustrate the directional adjustments needed for each of the four features in the sample.



```
concise_ssd
```

```
{'Class 0': {'Feature 1': {'positive': [],
   'neutral': ['H', 'M', 'L'],
   'negative': []},
  'Feature 2': {'positive': ['H'], 'neutral': ['M', 'L'], 'negative': []},
  'Feature 3': {'positive': ['L'], 'neutral': [], 'negative': ['H', 'M']},
  'Feature 4': {'positive': ['L'], 'neutral': [], 'negative': ['H', 'M']}},
 'Class 1': {'Feature 1': {'positive': [],
   'neutral': ['H', 'M', 'L'],
   'negative': []},
  'Feature 2': {'positive': [], 'neutral': ['H', 'M', 'L'], 'negative': []},
  'Feature 3': {'positive': ['M'], 'neutral': ['H', 'L'], 'negative': []},
  'Feature 4': {'positive': ['M'], 'neutral': ['L'], 'negative': ['H']}},
 'Class 2': {'Feature 1': {'positive': [],
   'neutral': ['H', 'M', 'L'],
   'negative': []},
  'Feature 2': {'positive': [], 'neutral': ['H', 'M', 'L'], 'negative': []},
  'Feature 3': {'positive': ['H'], 'neutral': [], 'negative': ['M', 'L']},
  'Feature 4': {'positive': ['H'], 'neutral': [], 'negative': ['M', 'L']}}}
```

(a) Concise SSD For Iris Dataset



```
conversion_table
```

```
{(0, 1): [['-'], ['-'], ['M'], ['M']],
 (0, 2): [['-'], ['-'], ['H'], ['H']],
 (1, 0): [['-'], ['H'], ['L'], ['L']],
 (1, 2): [['-'], ['-'], ['H'], ['H']],
 (2, 0): [['-'], ['H'], ['L'], ['L']],
 (2, 1): [['-'], ['-'], ['M'], ['M']]}
```

(b) Conversion Table For Iris Dataset

Figure 8: Concise SSD and Conversion Table

The feature analysis for evasion begins with the categorization of features based on their influence levels - Low (L), Medium (M), and High (H), as per Equation 1. Following this, we progress to the initialization of the SHAP Summary based on Equation 3, which involves segregating SHAP values into three distinct categories: positive (P), neutral (N_T), and negative (N) based on the criteria detailed in Equation 2. This classification is pivotal in understanding the directional influence of each feature.

Once the SHAP summary is constructed, the subsequent step involves generating a concise summary. This is achieved by iterating through each class and feature, allowing us to categorize the impact of features within each class. To accomplish this, we analyze the occurrence counts of impact categories for each class and feature, considering the occurrence of these impact categories within the SHAP categories (Equation 4 and 5). Illustrating the process with an example: for a specific class and feature, if the impact category 'H' occurs 4 times in the 'positive' SHAP category and 2 times in the 'negative' SHAP category, we assign the impact category 'H' to the 'positive' category in the refined summary, prioritizing the higher occurrence within that particular SHAP category. This approach provides a balanced representation of feature impacts by averaging effects across the entire dataset, fostering a more robust understanding of relationships between feature values and their corresponding impact categories.

In addition to analyzing feature impact, we create a conversion table mapping the impact of each feature on the original class to its potential impact on the target class given a change in feature value. Initially, an empty conversion table is initialized, and possible class conversions are determined based on the number of unique classes present in the dataset (Equation 6). For each class conversion, we iterate through features and consider impact categories within both the original and target classes. By comparing these impact categories, we identify the direction in which the feature's value should be modified (Equation 7, 8 and 9).

B Algorithms

B.1 Evasion attack

In this section we provide a detailed description of the evasion attack algorithm employed in our research. This algorithm operates in a black-box setting, relying solely on the model's output predictions to guide the strategy. The iterative process aims to modify the features of the input sample, ultimately crossing the decision boundary into the target class.

Algorithm 1 Evasion Attack Strategy

```
Input: $x_{org}, c_{from}, c_{to}$
Output: $x_{adv}, success$
Data: $target_{model}, d_{max}, conversion_{table}, T_{low}, T_{high}$
$x_{adv} \leftarrow$ copy of $x_{org}$
$best_{adv} \leftarrow$ copy of $x_{org}$
$d_{least} \leftarrow 1$
$conversion_{rules} \leftarrow conversion_{table}[(c_{from}, c_{to})]$
for all feature in $x_{org}$ do
    Categorize $feature_{val}$ based on $T_{low}$ and $T_{high}$
    for all $conversion_{category}$ in $conversion_{rules}$[feature] do
        $x_{temp} \leftarrow$ copy of $x_{adv}$
        if $conversion_{category}$ is '-' or the category equals $conversion_{category}$ then
            continue
        end if
        Modify $x_{adv}$ from category to $conversion_{category}$
        Clip modified feature value to [0, 1]
        if $target_{model}$.predict($x_{adv}$) == $c_{to}$ then
            $d_{new}$ = distance($x_{adv}, x_{org}$)
            if $d_{new} < d_{least}$ then
                $best_{adv} \leftarrow$ copy of $x_{adv}$
                $d_{least} \leftarrow d_{new}$
            end if
        end if
    end for
end for
if $best_{adv} \neq x_{org}$ then
    return $best_{adv}$, True
else
    return $x_{adv}$, False
end if
```

B.2 Optimal Epsilon

In this section we provide an algorithm to obtain an Optimal Epsilon, that determines the smallest epsilon, ($\epsilon_{\text{optimal}}$) required for creating impactful adversarial samples. It employs a refined evasion attack approach, utilizing a binary search loop to iteratively narrow down epsilon ranges, ensuring the generation of effective adversarial samples with minimal perturbation.

Algorithm 2 Optimal Epsilon

```
Input: $x_{\text{org}}, c_{from}, c_{to}$
Output: $x_{\text{adv}}, d_{least}, \epsilon_{\text{optimal}}, success$
Data: $target_{model}, conversion_{table}, T_{low}, T_{high}, tolerance$
$\epsilon_{\text{low}} \leftarrow 0$
$\epsilon_{\text{high}} \leftarrow 0.5$
$x_{\text{adv}} \leftarrow$ copy of $x_{\text{org}}$
$best_{\text{adv}} \leftarrow$ copy of $x_{\text{org}}$
$d_{least} \leftarrow 1$
$conversion_{rules} \leftarrow conversion_{table}[(c_{from}, c_{to})]$
$\epsilon_{\text{optimal}} \leftarrow 1$
while $(\epsilon_{\text{high}} - \epsilon_{\text{low}}) >$ tolerance do
  $\epsilon_{\text{mid}} \leftarrow (\epsilon_{\text{low}} + \epsilon_{\text{high}})/2$
  $x_{\text{adv}} \leftarrow$ copy of $x_{\text{org}}$
  for all feature in $x_{\text{org}}$ do
    Categorize $feature_{val}$ based on $T_{low}$ and $T_{high}$
    for all $conversion_{category}$ in $conversion_{rules}[feature]$ do
      if $conversion_{category}$ is '-' or the category equals $conversion_{category}$ then
        continue
      end if
      Modify $x_{\text{adv}}$ from category to $conversion_{category}$
      Clip modified feature value to [0, 1]
      if $target_{model}$.predict($x_{\text{adv}}$) == $c_{to}$ then
        $d_{new}$ = distance($x_{\text{adv}}, x_{\text{org}}$)
        if $d_{new} < d_{least}$ then
          $best_{\text{adv}} \leftarrow$ copy of $x_{\text{adv}}$
          $d_{least} \leftarrow d_{new}$
          $\epsilon_{\text{optimal}} \leftarrow \epsilon_{\text{mid}}$
        end if
      end if
    end for
  end for
  if $(best_{\text{adv}} \neq x_{\text{org}})$ then
    $\epsilon_{\text{high}} \leftarrow \epsilon_{\text{mid}}$
  else
    $\epsilon_{\text{low}} \leftarrow \epsilon_{\text{mid}}$
  end if
end while
if $(best_{\text{adv}} \neq x_{\text{org}})$ then
  return $best_{\text{adv}}, d_{least}, \epsilon_{\text{optimal}}$, True
else
  return $x_{\text{adv}}, d_{least}, \epsilon_{\text{optimal}}$, False
end if
```