# Accelerated Adversarial Attack Generation and Enhanced Decision Insight

N.K.Y.S Kumarasiri
*Faculty of Information Technology*
*University of Moratuwa*
Moratuwa, Sri Lanka
yashmi110@gmail.com

S.C. Premaratne
*Faculty of Information Technology*
*University of Moratuwa*
Moratuwa, Sri Lanka
samindap@uom.lk

W.M.R.M Wijesuriya
*Faculty of Information Technology*
*University of Moratuwa*
Moratuwa, Sri Lanka
wijesuriyar@uom.lk

2023 8th International Conference on Information Technology Research (ICITR) | 979-8-3503-5950-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICITR61062.2023.10382716

***Abstract*— Adversarial Attack is a rapidly growing field that studies how intentionally crafted inputs can fool machine learning models. This can have severe implications for the security of machine learning systems, as it can allow attackers to bypass security measures and cause the system to malfunction. Finding solutions for these attacks involves creating specific attack scenarios using a particular dataset and training a model based on that dataset. Adversarial attacks on a trained model can significantly reduce accuracy by manipulating the decision boundary, causing instances initially classified correctly to be misclassified. This alteration results in a notable decline in the model's ability to classify instances after an attack accurately. The above process helps us develop strategies to defend against these attacks. However, a significant challenge arises because generating these attack scenarios for a specific dataset is time-consuming. Moreover, the disparity between the model's prediction outcomes before and after the attack tends to lack clear interpretability. In both above limitations, the common limiting factor is time. The time it takes to devise a solution is crucial because the longer it takes, the more opportunity an attacker has to cause harm in real-world situations. In this paper, we propose two approaches to address the above gaps: minimizing the time required for attack generation using data augmentation and understanding the effects of an attack on the model's decision-making process by generating more interpretable descriptions. We show that description can be used to gain insights into how an attack affects the model's decision-making process by identifying the most critical features for the model's prediction before and after the attack. Our work can potentially improve the security of machine learning systems by making it more difficult for attackers to generate effective attacks.**

***Keywords*— *Adversarial Machine Learning, Adversarial Attack, Explainable AI***

## I. Introduction

Adversarial machine learning holds a pivotal role within the realm of machine learning. Over the last decade, a significant amount of research has been dedicated to adversarial machine learning. Adversarial machine learning involves creating algorithms that can defend against advanced attacks and studying the strengths and weaknesses of these attacks [1]. In the context of adversarial attacks, we can change how a model predicts by adding slight, hard-to-detect changes to clean data. Even though these changes are minimal, and humans can't notice them, they can affect the model's output [2]. This knowledge empowers us to fortify models against real-world threats and enhance their robustness. For example, consider a self-driving car that relies on machine learning to recognize road signs and navigate. Adversarial machine learning is essential in this scenario. Picture an attacker making small, unnoticeable changes to the appearance of road signs. These changes might not be visible to humans but could confuse the car's AI. If the vehicle isn't prepared for such attacks, it could mistake a stop sign for a yield sign, potentially causing accidents. Adversarial machine learning is like a test where researchers intentionally try to confuse the car's AI system by making subtle changes to road signs. This helps them understand the car's weaknesses and how it might respond incorrectly. By doing this, they can figure out ways to make the model more resistant to these tricks.

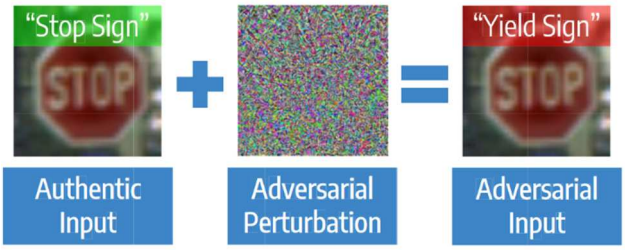


Fig. 1. Visual representation of how adversaries fool the model[3].

From the research standpoint, while significant efforts have been dedicated to this field, minimal attention has been directed toward the time required for generating attacks. As the chosen data subset increases, the time taken to execute attacks on the dataset also increases. In real-world situations, quick responses to attacks are crucial. Exploring the connection between attack generation time and data subset size could offer insights into improving defenses against such attacks. Filling this gap would enhance our understanding of adversarial challenges and lead to better defensive approaches.

The other limitation we are going to address in this study is that individuals working with the machine learning models lack insight into the decision-making process of them after an adversarial attack. This means they can't know how the model decides a particular record's exact class label after an attack. This is primarily due to the models' limited explainability and interpretability [4]. By exploring how decisions are made after attacks, it can help create models that are easier to understand. This can benefit data scientists and analysts, enabling them to verify model results even when attacks happen. Bridging this gap could lead to more informed and confident decision-making, which is pivotal in domains where accurate and reliable model predictions are essential, such as healthcare, finance, and autonomous systems.

This study aimed to tackle the mentioned limitations in two ways. First, we used data augmentation techniques [5] to speed up adversarial attack generation. Secondly, we have introduced an innovative approach using explainable AI, particularly LIME (Local Interpretable Model-Agnostic Explanations) [6], to create explanations that shed light on the decision-making process before and after attacks. This

significantly improves the models' interpretability and clarity. We conducted this experiment using the NSL-KDD dataset and the Europe Credit Card dataset.

## II. Literature Review

### A. Adversarial Attacks

For over a decade, researchers have dedicated their efforts to studying adversarial attacks [7]. In adversarial machine learning, attackers aim to deceive systems into making incorrect decisions. This manipulation occurs either during the inference phase, called evasion attack, or during the training phase, called poisoning attack [8]. We can better understand adversarial attacks' potential effects on machine learning systems by investigating adversarial attacks. These outcomes primarily hinge on factors like the attacker's expertise, the targeted area of attack, their chosen attack strategy, and the scope of the attack [9]. In other words, we can call it adversarial attack taxonomy, a system for classifying different adversarial attacks.

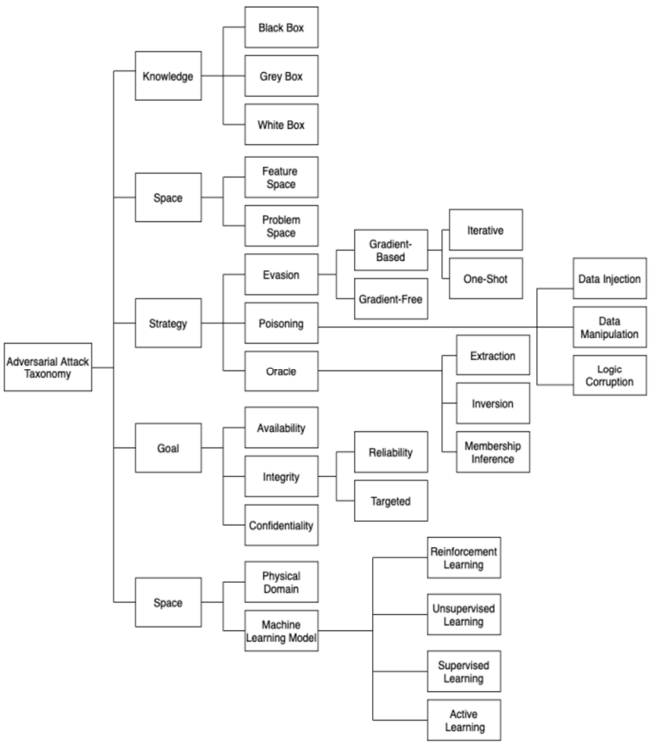


Fig. 2. Adversarial Attack Taxonomy*[10]*.

### B. Time Consumption of Adversarial Attack Generation

Creating attacks is time-consuming because it's similar to solving a complex puzzle. We tried to find small changes in the input that trick the model. This process requires many calculations and tests, and it becomes more challenging if the model has solid defenses or your tools (hardware) aren't very fast. It's comparable to discovering a hidden code, and the more you want to be successful, the more time it generally requires. Researchers of the study [11] conducted experiments by progressively extending the delay in adversarial training across different numbers of iterations (0, 10k, 20k, and 40k). Initially, they exclusively employed clean examples, gradually incorporating adversarial examples afterward. Their observations indicate that longer delays contributed to a potential decline of up to 4% in accuracy for more potent attacks. The selected delay, which has 10k iterations, reveals the correlation between the increase in delay and the extended time required for training the model.

In the study of [12], researchers thoroughly examined existing methods to improve defenses against adversarial attacks. They introduced a new way to categorize these methods, which is a significant contribution. They also explored the challenges of applying these techniques broadly and collected benchmark results for comparison. The study emphasized that generating adversarial attacks takes more time with larger datasets. Overall, this research enhances our understanding of adversarial defense strategies.

### C. Interpretability of the decision-making process of attacked models

Using non-transparent machine learning in critical roles like assistive and service robots is risky. Lack of explanations for their decisions could lead to harmful consequences. So, explainability is crucial for these tasks [4].

The study [13] introduces a novel method to explain misclassifications in datasets tested on an Intrusion Detection system. Using an innovative adversarial approach, it identifies small changes needed to correct misclassifications and visualizes the key features causing them. It's evaluated with Linear and Multilayer Perceptron classifiers, yielding insightful graph explanations that align with expert knowledge. Notably, patterns emerge: regular connections with short duration and low login success are misclassified as attacks, while specific attributes lead to reversed misclassifications. This research provides a valuable visualizing tool for understanding and addressing misclassification challenges in complex datasets.

### D. Europe Credit Card Dataset

The dataset consists of credit card transactions conducted by European cardholders during September 2013. It was compiled through a collaboration between the Machine Learning Community and ULB for fraud detection and extensive data analysis. The dataset spans two days and comprises 284,315 transactions, encompassing both valid and fraudulent purchases. Notably, the dataset is clean and devoid of duplicate entries, significant outliers, or missing values, thereby eliminating the need for data cleaning [12]. However, it's important to note that the credit card dataset lacks certain numerical attributes due to privacy constraints set by the data issuer organization. Owing to security constraints, the dataset's feature names are not accessible.

### E. NSL KDD Dataset

The KDD CUP 99 dataset was produced as a result of decades of study in the Network Intrusion Detection sector to address its problems and produce the NSL-KDD dataset. This makes NSL-KDD the most recent dataset. The NSL-KDD dataset offers 41 variables that represent various aspects of network flow. We had to convert all the qualitative values to numeric values using one-hot encoding. The 42nd attribute includes information about 39 attacks, each of which is divided into five classifications. One normal class and four attack classes make up these five classes. Denial of Service (DoS), Root to Local (R2L), Probe, and User to Root (U2R) are the four classes of attacks. This realism allows researchers

to simulate real-world scenarios, making it an ideal environment to assess the impact of adversarial attacks and study decision-making processes before and after an attack. This dataset is a widely used benchmark dataset for network intrusion detection. It consists of approximately 4.9 million records, making it a substantial and comprehensive dataset for research and evaluation. In our results chapter, we emphasize the analysis of features within the NSL-KDD dataset. To facilitate a deeper understanding, we have presented some of these features in the table below [14].

TABLE I. NSL-KDD DATASET DESCRIPTION

| Feature ID | Feature Name |
|---|---|
| 1 | duration |
| 2 | protocol_type |
| 3 | service |
| 4 | flag |
| 5 | src_bytes |
| 6 | dst_bytes |
| 7 | land |
| 8 | wrong_fragment |
| 9 | urgent |
| 10 | hot |
| 11 | num_failed_logins |
| 12 | logged_in |
| 13 | num_compromised |

## III. METHODOLOGY

After conducting the literature review, it's evident that while there have been studies in the field of adversarial machine learning, limited attention has been given to the following areas.

- Minimizing the time required for attack generation.
- Understanding the effects on the model's decision-making process after an attack.

In our approach, we begin by obtaining the NSL-KDD and Europe Credit Card datasets and training a model using a random forest classifier after completing standard preprocessing. To bridge the identified research gaps, we devised two distinct approaches. To tackle the challenge of reducing attack generation time, we leveraged the potential of data augmentation. By implementing this technique, we effectively streamlined the process of generating adversarial attacks, thus addressing a crucial aspect that had been previously underemphasized.

Secondly, to gain insights into the model's decision-making process, LIME is employed to specific one instance of the dataset. LIME helps explain the predictions by highlighting the contribution and weight of each feature. Our approach involves implementing and testing three types of adversarial attacks: Zoo and Elastic Net attacks. After generating an attack, the selected machine learning model (in this case, random forest classifier) is retrained using the attacked dataset. LIME is applied again to the same instance we previously considered to compare the impact of features before and after the attack. This involves analyzing the weight and contribution values of the feature towards the model's decision-making process. A description is generated based on the contribution impact value of features to predict a specific outcome.

### *A. Minimizing the time required for attack generation*

When dealing with the NSL-KDD dataset, its substantial size of over 4 million records poses challenges for research purposes. First, we opt for a representative random subset from the dataset. Additionally, we introduce an "attack portion" to specify the number of samples reserved for generating attacks. If the chosen data subset is larger than the predefined "attack_portion," the attack is executed on this specified attack portion.

Then, to create augmented data, a list comprehension repeatedly calls the augment data function on the targeted attack portion, concatenating the outcomes along the first axis (rows). Within data augmentation, shuffling is consistently conducted, resulting in the same shuffling pattern every time the function is employed with identical input data. Random noise values, characterized by a mean of 0 and a standard deviation of 0.05, are generated and added element-wise to the shuffled data.

Our method ensures that the augmented data contains at least the same number of samples as the initially defined data portion. If the number of rows in the initial data portion isn't greater than the attack portion, the entire data portion becomes the focus of the attack. In simpler terms, the logic can be explained as follows:

*Data = Get a Random small portion of the dataset*
***Repeat***
*data_parmuted = shuffle (data)*
*data_noise = Add Random noise to the data_parmuted*
***Until*** *the original dataset size*

### *B. Exploring Model Interpretability after the Attack*

This part of the approach serves as a toolkit for unraveling the inner workings of machine learning models. This approach employs a technique called LIME. This technique aids in comprehending the reasons behind a model's predictions. Several methods are designed to dissect and compare the significance of model features before and after an adversarial attack.

First, to get an idea about how individual features influence the model's predictions, our approach generates a list of sorted features based on their impact on the model's decision. Then, a detailed story is crafted, explaining how an attack affected the importance of different features.

Then, the system compares the outputs before and after the attack and carefully examines whether certain factors gained or lost significance due to the attack. Finally, it creates informative descriptions that help us understand why these changes occurred and what they mean for the model's decisions. The visual representation of the whole process is shown in Fig. 3.

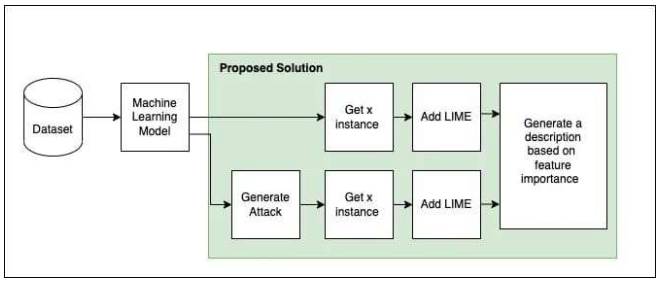

Fig. 3. Overview of the proposed solution.

In simpler terms, the logic can be explained as follows:

*Instance = get a single instance of attack dataset*
***While** (Iterate every single feature)*
*Compare feature instance with the original dataset.*
***End While***

## IV. Experimental Results

### A. Streamlining Attack Generation Time through Data Augmentation

We tested the findings by comparing attack generation times before and after applying the data augmentation technique. Earlier research efforts have focused on specific attacks within smaller data subsets. Our approach involves selecting a data subset, evaluating and attacking a sub-portion, and then expanding this data until it aligns with the primary dataset/ data subset. We assessed the time taken both prior to and following the adoption of the data augmentation strategy. Our assessment encompassed two datasets (European credit card and NSL-KDD), two types of attacks (Zoo and Elastic Net attacks), and a single machine learning model, the Random Forest Classifier.

#### 1) Testing with Zoo Attack

The tables provided showcase the time taken before and after implementing the augmentation technique, along with the corresponding dataset and the specific portions of data that were taken into account.

TABLE II. BEFORE ATTACK

| Dataset | Attacked Portion | Time Consumed (minutes) | Accuracy |
|---|---|---|---|
| Europe Credit Card | 472 rows | 100 | 62.5% |
| NSL-KDD | 4,019,987 rows | 300 | 54.3% |

TABLE III. AFTER ATTACK

| Dataset | Attacked Portion | Time Consumed (minutes) | Accuracy |
|---|---|---|---|
| Europe Credit Card | 472 rows | 0.3 | 42.5% |
| NSL-KDD | 4,019,987 rows | 1.17 | 32.2% |

#### 2) Testing with Elastic Net Attack

The tables provided showcase the time taken before and after implementing the augmentation technique, along with the corresponding dataset and the specific portions of data that were taken into account.

TABLE IV. BEFORE ATTACK

| Dataset | Attacked Portion | Time Consumed (minutes) | Accuracy |
|---|---|---|---|
| Europe Credit Card | 472 rows | 97 | 67% |
| NSL-KDD | 4,019,987 rows | 270 | 42.6% |

TABLE V. AFTER ATTACK

| Dataset | Attacked Portion | Time Consumed (minutes) | Accuracy |
|---|---|---|---|
| Europe Credit Card | 472 rows | 0.6 | 55.2% |
| NSL-KDD | 4,019,987 rows | 1.05 | 32.1% |

Fig. 4. Is a self-composed visual representation of the time required for attack generation before and after data augmentation. For this analysis, we selected portions of 0.1, 0.2, 0.3, and 0.4 from the Europe Credit Card dataset and subjected them to a Zoo attack. The results indicate that before data augmentation, the time taken significantly increases with the attacked portion size. However, after augmenting data, there is only a slight change in time consumption, which remains relatively constant compared to the time before data augmentation.

After subjecting the dataset to attacks, we expect the model to experience an increase in misclassifications, thereby leading to a decrease in overall accuracy. In TABLE II, III, IV, and V, we compare the accuracy of the attacked model before implementing data augmentation, which is the conventional approach adopted by most researchers, and after applying our data augmentation technique. Analyzing the obtained accuracies, it becomes evident that our approach significantly reduces accuracy. This outcome aligns with our ultimate goal and demonstrates the effectiveness of our approach.

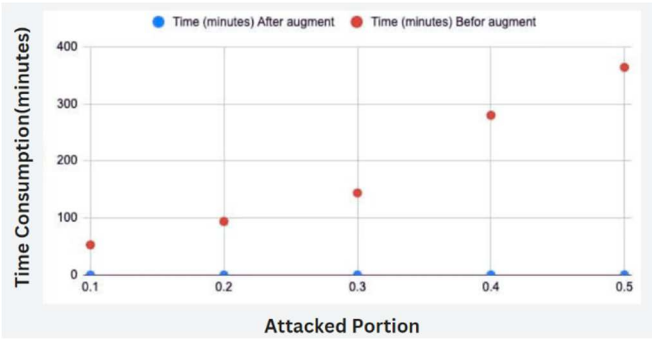

Fig. 4. Attack Portion vs Time Consumption.

### B. Enhancing Interpretability

An instance from the NSL KDD dataset has been considered for explanation. Fig. 5. shows the initial step of the process. Before generating an attack, an instance is selected from the dataset. The LIME method is then applied to this instance, resulting in an array comprising all dataset features accompanied by a conditional value and additional values.


Fig. 5. Feature description before the attack

As shown in Fig. 6, after the attack, the identical instance is used to generate a feature description using the same process as described above.

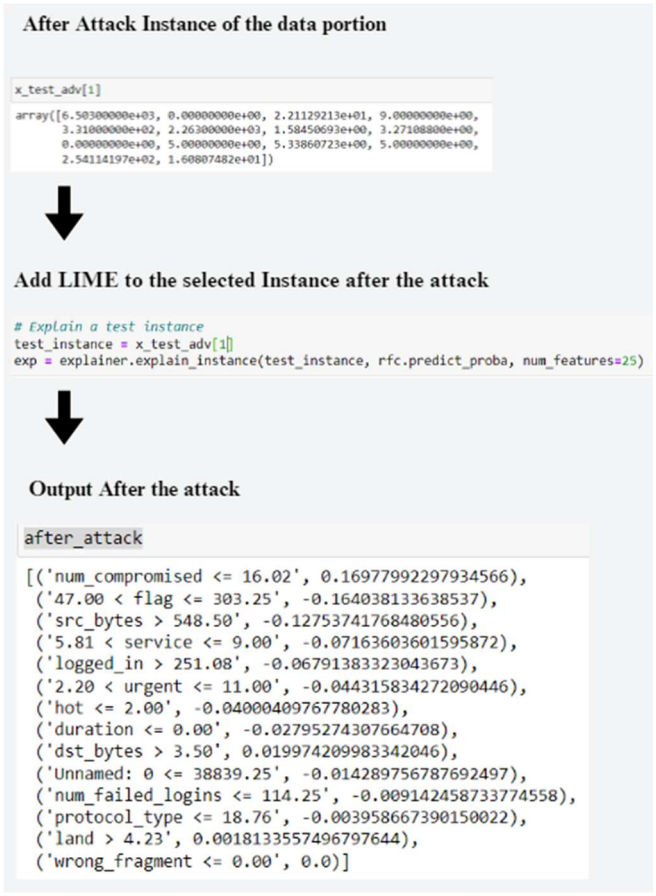


Fig. 6. Feature description after the attack

If we consider a particular value from the array, the conditional value is an approximation of the behavior of the original complex model around the specific instance. The value next to the condition is the contributing impact to predict a particular outcome. Suppose we consider that impact value separately. There can be six different scenarios. For example, let's consider the condition of the src_bytes feature as "src_bytes > 466.50". Assume the condition is constant.

TABLE VI. SAMPLE DESCRIPTION CONSIDER ONE FEATURE

| **src_bytes > 466.50** | | |
|---|---|---|
| ***Scenarios*** | ***Impact value*** | ***Description*** |
| Considering the sign (Positive or Negative) | -0.5480 | The feature is associated with a lower likelihood of the predicted class/ Feature contributing less likely to predict a specific outcome |
| | 0.5480 | The feature is associated with a higher likelihood of the predicted class/ Feature contributing more likely to predict a specific outcome |
| Considering magnitudes | -0.1480 | A not so stronger **NEGATIVE** Impact / Feature is not so strongly associated with a **LOWER** likelihood of the predicted outcome |
| | - 0.9480 | A stronger **NEGATIVE** Impact / Feature is more strongly associated with a **LOWER** likelihood of the predicted outcome |
| | 0.1480 | A not so stronger **POSITIVE** Impact / Feature is not so strongly associated with a **HIGHER** likelihood of the predicted outcome |
| | 0.9480 | A stronger **POSITIVE** Impact / Feature is more strongly associated with a **HIGHER** likelihood of the predicted outcome |

Finally, for the selected instance, we analyze the importance of features before and after the attack. For the purpose of explanation, let's consider four features in the NSL KDD dataset: land, dst_bytes, num_failed_logins, and wrong_fragment. So before the attack, the conditional values and impact values we are getting are like below.

*land <= 0.00', -0.021366161962117474*
*dst_bytes <= 0.00', -0.0197892368681665*
*num_failed_logins <= 89.00', -0.02628401959032776*
*wrong_fragment <= 0.00', 0.0076828529290236735*

When the model makes a decision for this specific instance, the significance of features is as follows.

*wrong_fragment > dst_bytes > land > num_failed_logins*

Following the attack, the obtained conditional and impact values are shown below.

*land > 4.23', 0.0018133557496797644*
*dst_bytes > 3.50', 0.019974209983342046*

*num_failed_logins <= 114.25', -0.009142458733774558*
*wrong_fragment <= 0.00', 0.0*

When the model makes a decision for the same specific instance after the attack, the significance of features is as follows.

*dst_bytes > land >wrong_fragment > num_failed_logins*

Utilizing the changes of significance before and after the attack, the generated description is finally as follows.
*"The adversarial attack has manipulated the model into incorrectly perceiving the importance of dst_bytes, land as increased and the importance of wrong_fragment as decreased. The significance of num_failed_logins remains the same even after the attack."*

## V. Discussion

The study's outcomes shed light on existing gaps in research concerning adversarial attacks. This investigation explicitly addresses two key gaps. First, there's a substantial increase in attack generation time when the attack portion is larger. This issue is mitigated by implementing a data augmentation technique, which significantly reduces the required time, as evidenced by the results. While numerous researchers have acknowledged the time-intensive nature of generating adversarial attacks, a notable gap exists in addressing and proposing solutions to this issue. Consequently, a comparative analysis of existing solutions remains unfeasible, as the identified time consumption challenge has not been adequately tackled in the current body of research.

Second, a gap exists in understanding how the decision-making process of a machine learning model changes post-attack. Upon comparing our proposed solution for this particular gap with that of [13], it is observed that the latter includes visual representations of false positive and false negative values utilizing the difference between actual samples and modified samples. These illustrations depict the outcomes following model training on the attacked NSL-KDD99 dataset using a linear classifier.

Our approach utilizes LIME to create clear descriptions, as outlined in the methodology. These descriptions showcase how attacks distort the model by altering the significance of features, resulting in inaccurate classifications. The notable part of our methodology lies in its ability to make these descriptions easily understandable for a broad audience.

## VI. Conclusion

By leveraging adversarial attack generation, we enhance the robustness of our systems by subjecting our models to simulated attacks to identify vulnerabilities and weak points systematically. Through this process, we gain valuable insights into potential pitfalls, enabling us to fortify our models against adversarial manipulation.

As for future works, to specifically address models that may have suffered significant drops in performance due to previous attacks, we can incorporate adversarial training, which means training the model on both clean and adversarial examples to promote robust feature learning. Furthermore, we can emphasize robust feature engineering, which means selecting features that are less susceptible to adversarial manipulation. This comprehensive approach strengthens our models against potential threats and helps improve the accuracy of models that have experienced performance degradation. Proactively addressing vulnerabilities and implementing these strategies can ensure more reliable and accurate model outcomes.

## VII. References


[1] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, "Adversarial machine learning".

[2] N. Liu, M. Du, R. Guo, H. Liu, and X. Hu, "Adversarial Attacks and Defenses: An Interpretation Perspective".

[3] F. Carrara, F. Falchi, G. Amato, R. Becarelli, and R. Caldelli, "Detecting Adversarial Inputs by Looking in the black box".

[4] V. Dutta and T. Zielińska, "An Adversarial Explainable Artificial Intelligence (XAI) Based Approach for Action Forecasting," *JAMRIS*, pp. 3–10, Mar. 2021, doi: 10.14313/JAMRIS/4-2020/38.

[5] A. Antoniou, A. Storkey, and H. Edwards, "Data Augmentation Generative Adversarial Networks." arXiv, Mar. 21, 2018. Accessed: Dec. 02, 2023. [Online]. Available: http://arxiv.org/abs/1711.04340

[6] D. Garreau, "Explaining the Explainer: A First Theoretical Analysis of LIME".

[7] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, Dec. 2018, doi: 10.1016/j.patcog.2018.07.023.

[8] L. Muñoz-González *et al.*, "Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization." arXiv, Aug. 29, 2017. Accessed: Dec. 02, 2023. [Online]. Available: http://arxiv.org/abs/1708.08689

[9] B. Biggio *et al.*, "Evasion Attacks against Machine Learning at Test Time," in *Advanced Information Systems Engineering*, vol. 7908, C. Salinesi, M. C. Norrie, and Ó. Pastor, Eds., in Lecture Notes in Computer Science, vol. 7908. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 387–402. doi: 10.1007/978-3-642-40994-3_25.

[10] O. Ibitoye, R. Abou-Khamis, M. el Shehaby, A. Matrawy, and M. O. Shafiq, "The Threat of Adversarial Attacks on Machine Learning in Network Security -- A Survey." arXiv, Mar. 21, 2023. Accessed: Dec. 02, 2023. [Online]. Available: http://arxiv.org/abs/1911.02621

[11] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale." arXiv, Feb. 10, 2017. Accessed: Dec. 02, 2023. [Online]. Available: http://arxiv.org/abs/1611.01236

[12] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent Advances in Adversarial Training for Adversarial Robustness." arXiv, Apr. 20, 2021. Accessed: Dec. 02, 2023. [Online]. Available: http://arxiv.org/abs/2102.01356

[13] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An Adversarial Approach for Explainable AI in Intrusion Detection Systems." arXiv, Nov. 28, 2018. Accessed: Dec. 02, 2023. [Online]. Available: http://arxiv.org/abs/1811.11705

[14] A. KumarShrivas and A. Kumar Dewangan, "An Ensemble Model for Classification of Attacks with Feature Selection based on KDD99 and NSL-KDD Data Set," *IJCA*, vol. 99, no. 15, pp. 8–13, Aug. 2014, doi: 10.5120/17447-5392.