Data Poisoning in Learning Systems

Athish Ck
Presenter

Exploring the detection and prevention of label flipping attacks in credit card fraud detection systems to ensure data integrity and security.

Fraud Detection Vulnerabilities

Addressing vulnerabilities in fraud detection models

- **Data Poisoning Threat**

  Fraud detection models are exposed to data poisoning, leading to misclassification.

- **Misclassification Impact**

  Increased fraudulent transactions are incorrectly classified as legitimate, worsening fraud cases.

- **Label Flipping Attack**

  Implementing a label flipping attack on the fraud detection dataset to evaluate vulnerabilities.

- **Detection Mechanism Development**

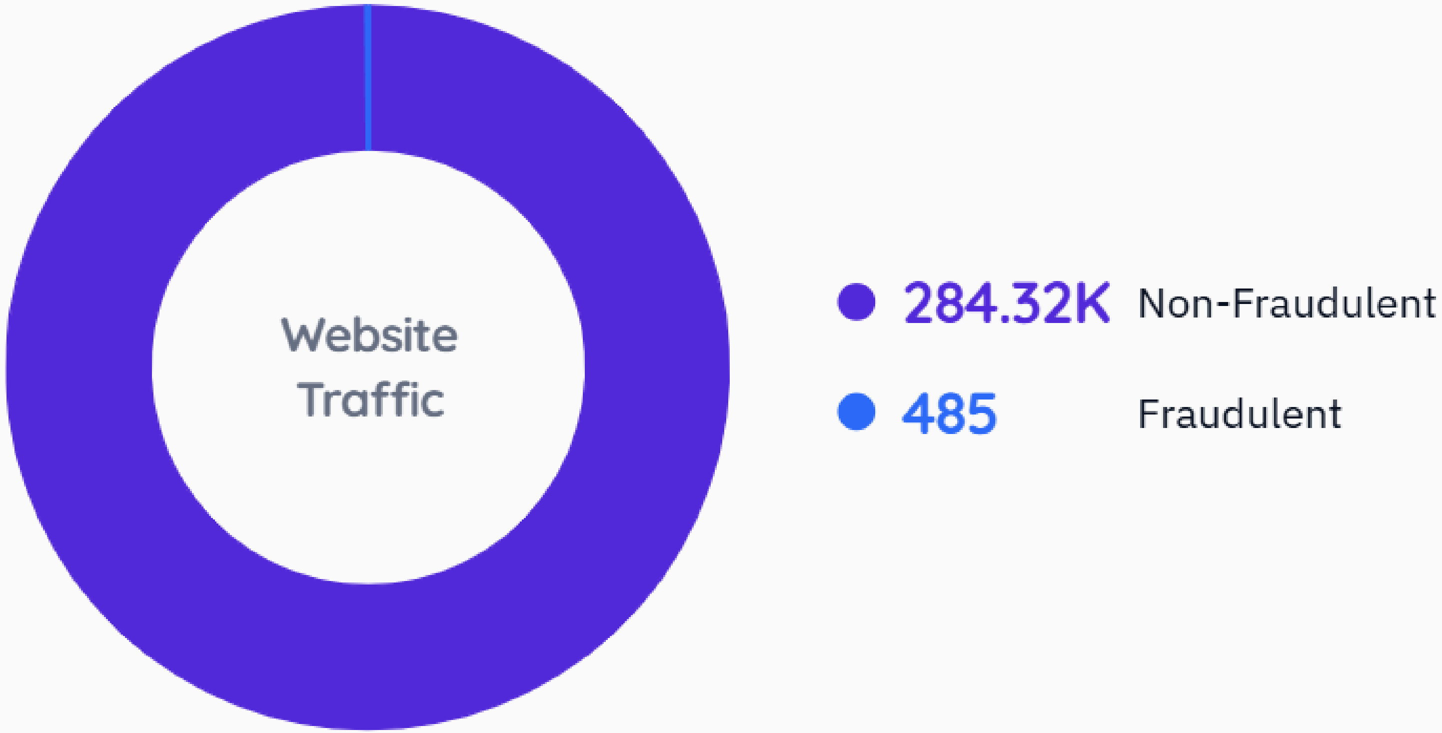
  Developing mechanisms to detect and identify poisoned data within the dataset.

- **Correction Techniques Application**

  Applying various correction techniques to improve fraud detection accuracy after attacks.

Credit Card Fraud Dataset Insights

Analysis of fraudulent transactions in dataset



Credit Card Fraud Dataset

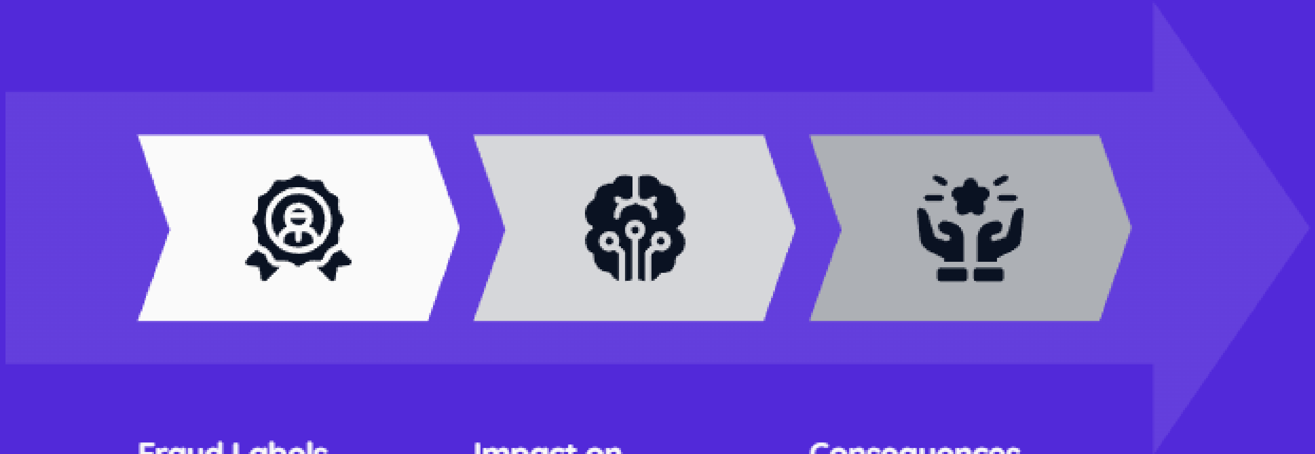
Label Flipping Attack Implementation

Understanding the mechanics and impact of label flipping

```
flip_percentage=0.5 #50% flipped

fraud_indices=dataf[dataf['Class']==1].index


#Selecting the fraud samples to filp
flip_count=int(len(fraud_indices)*flip_percentage)
flip_indices=np.random.choice(fraud_indices,flip_count,replace=False)

#Flip fraud to non fraud
dataf.loc[flip_indices,'Class']=0

print("flipped",flip_count)
print(dataf['Class'].value_counts())
```

✓ 0.0s

```
flipped 246
Class
0    284561
1       246
Name: count, dtype: int64
```



Fraud Labels Flipped

30% of fraud labels (1) are flipped to non-fraud (0), altering training data.

Impact on Model Training

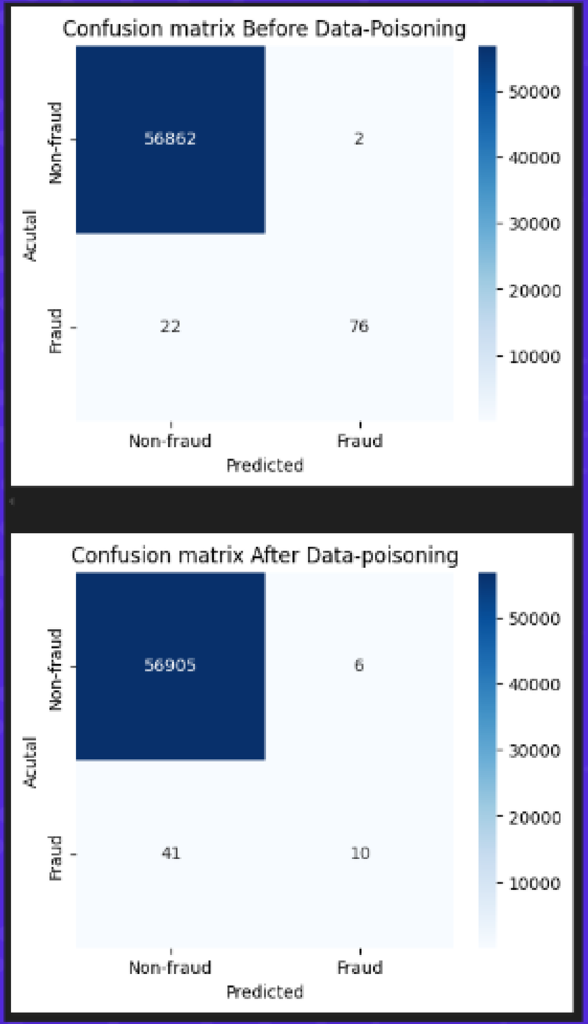
The label flipping leads to the model missing critical fraud cases during training.

Consequences of Attack

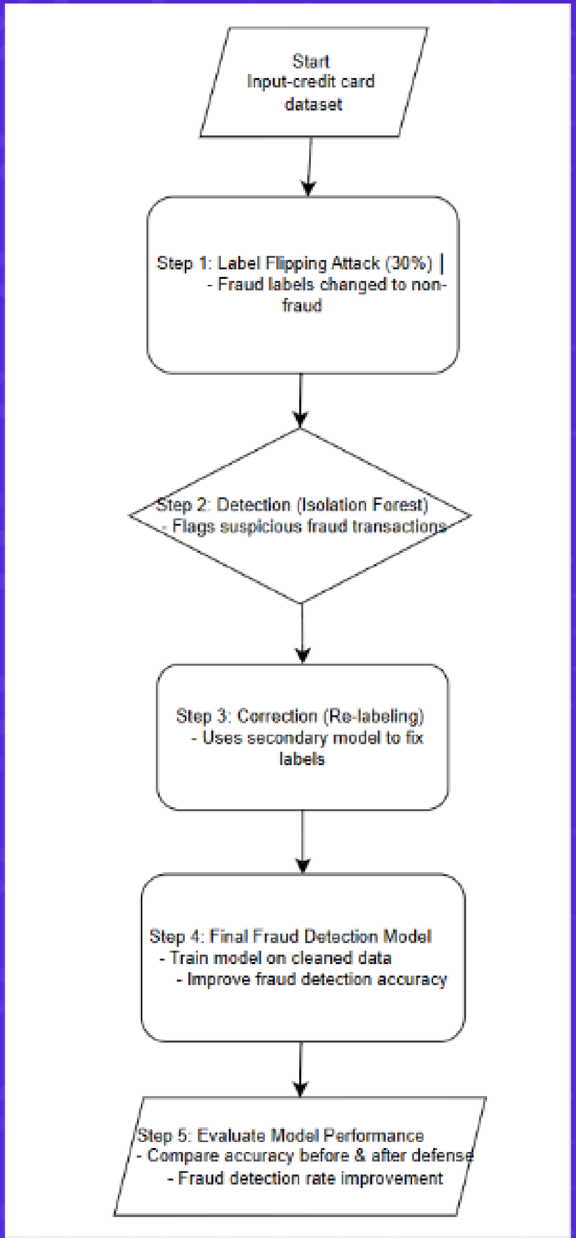
Increased likelihood of undetected fraudulent transactions due to misclassification.

Impact of Attack on Model Performance

Analysis of fraud detection before and after attack



Internal Fraud Detection System Data



Model Architecture: Defense Mechanism

Understanding the Fraud Detection Strategy

Step 1: Detection

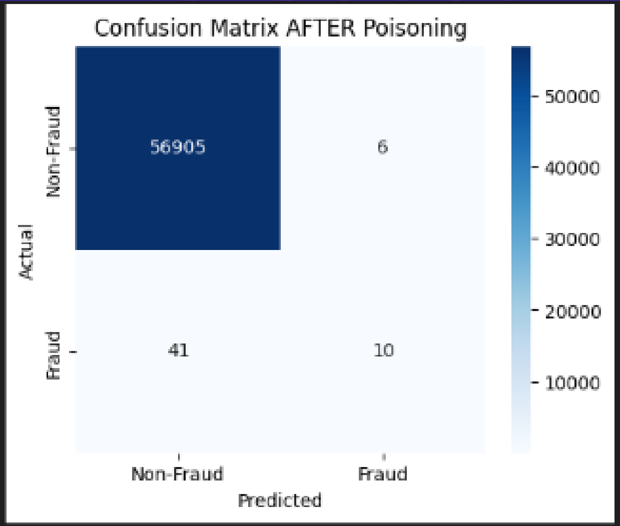
Utilize Isolation Forest to identify and flag suspicious transactions effectively.

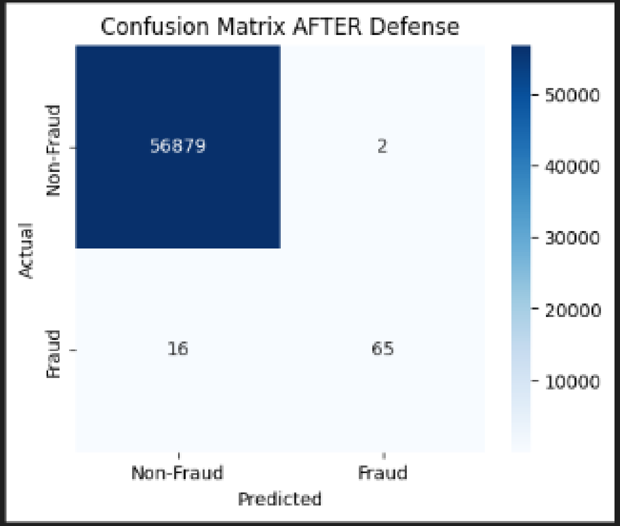
Step 2: Correction

Train a secondary model to relabel the samples that have been flagged as suspicious.

Step 3: Final Classification

Develop the fraud detection model using the cleaned data for accurate results.





Defense System Implementation Steps

A structured approach to enhancing data integrity

Step 1: Detect Poisoned Data

Utilized Isolation Forest to identify anomalies within data, ensuring integrity.

Step 2: Correct Poisoned Labels

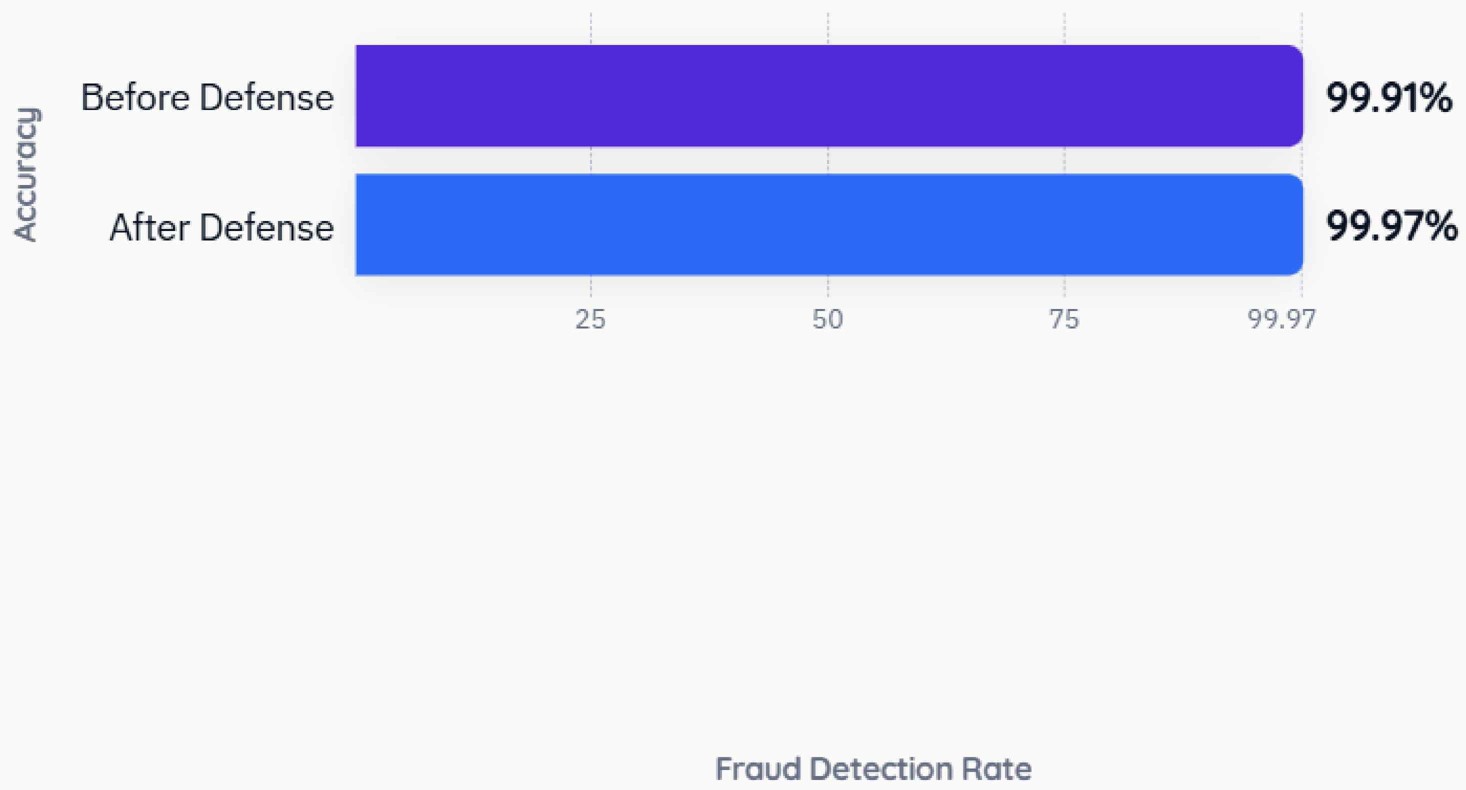
Reclassified fraudulent transactions using a clean model for accuracy in detection.

Step 3: Train Final Model

Retrained the fraud detection model post correction of poisoned labels for improved performance.

Results After Defense Improvements

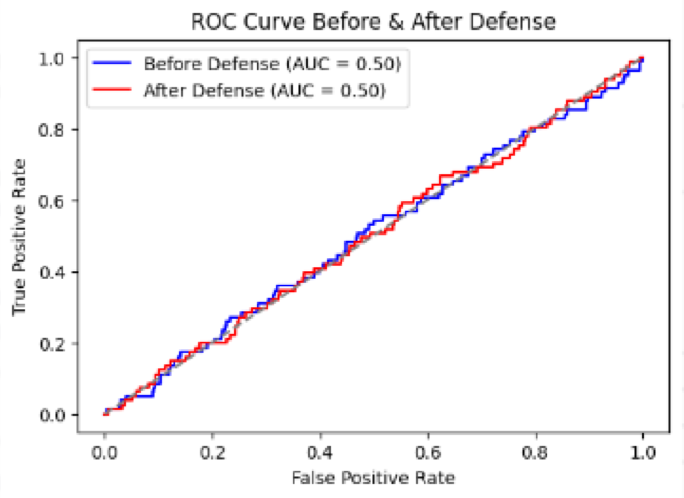
Significant advancements in detection metrics

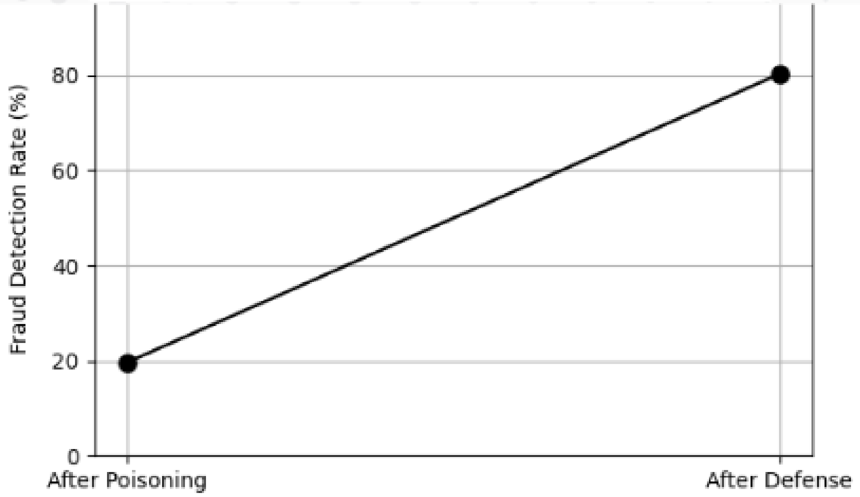


Source: Companies Market Cap

Conclusion and Future Directions

Insights from our findings and future initiatives





- **Mitigating Poisoning Attacks**

  Our innovative approach has demonstrated success in mitigating the effects of poisoning attacks on models.

- **Future Exploration of Adversarial Training**

  Future work will explore adversarial training methods to enhance defenses against model attacks.

- **Real-time Anomaly Detection**

  Implementing real-time anomaly detection systems in financial applications is an essential next step.

Source-code

Data-Poisoning link

https://colab.research.google.com/drive/1-eji3KqHQ9b7Z_7wiQ2z7GP-ZQAOelwl?usp=sharing