AI-Enhanced Retail Cybersecurity Threat Detection System

Arjit Jain
Indian Institute of Technology, Hyderabad
Kandi, Sangareddy
ai22btech11002@iith.ac.in

C Surya Saketh
Indian Institute of Technology, Hyderabad
Kandi, Sangareddy
ai22btech11005@iith.ac.in

Umanshiva Ladva
Indian Institute of Technology, Hyderabad
Kandi, Sangareddy
ai22btech11016@iith.ac.in

Mayank Parasramka
Indian Institute of Technology, Hyderabad
Kandi, Sangareddy
ai22btech11018@iith.ac.in

1. Abstract

This project aims to enhance cybersecurity for retail environments by developing an AI-powered system that focuses on protecting Point of Sale (POS) systems and network infrastructure. The system utilizes machine learning models to analyze data in real-time, detect anomalies, and classify potential security threats. By leveraging advanced data analysis techniques, the project provides an effective solution for identifying unusual activity and responding swiftly to potential breaches. The goal is to improve the security posture of retail businesses, minimizing the risk of data breaches and ensuring the smooth operation of POS systems. The system's performance is assessed based on its accuracy, efficiency, and ability to detect and mitigate security threats proactively.

2. Introduction

Cybersecurity is a critical aspect of modern business operations, especially in the retail sector, where protecting sensitive customer and transaction data is paramount. With the increasing use of digital technologies, cyberattacks have become more sophisticated and frequent, posing significant risks to businesses and consumers alike. Traditional security measures often struggle to keep up with the evolving tactics of cybercriminals, highlighting the need for advanced, adaptive solutions.

This project addresses the challenge of detecting and classifying cybersecurity threats in retail environments by leveraging artificial intelligence (AI). Specifically, it focuses on developing an AI-powered system that can analyze POS (Point-of-Sale) system logs and network traffic data to identify anomalies and potential threats. By using machine learning techniques, the project aims to enhance the ability of retailers to preemptively identify suspicious activities and respond effectively.

The primary objective of this project is to build and deploy a machine learning model capable of real-time threat detection and classification. This report outlines the methodology used, the design of the system, and the results obtained, contributing valuable insights into how AI can bolster cybersecurity in the retail industry.

3. JWT Authentication

JSON Web Tokens (JWT) is a widely adopted mechanism for stateless authentication. A JWT consists of three parts: the header, payload, and signature. The payload contains user-specific claims, while the signature ensures data integrity and authenticity. JWT is extensively used in modern web applications for secure communication between clients and servers.

3.1. How JWT Works

1. **Token Generation**: Upon successful login, the server generates a JWT containing user information and an expiration time, signed with a secret key.
2. **Token Validation**: Clients include the token in the `Authorization` header for subsequent requests. The server validates the token by verifying its signature and checking its expiration.
3. **Access Control**: Only valid tokens are granted access to protected resources.

3.2. Preventing SQL Injection with JWT

SQL injection is an attack where malicious SQL code is injected into inputs, exploiting insecure queries. JWT helps mitigate SQL injection in the following ways:

1. **Elimination of Repeated Database Queries**: Traditional session-based systems query the database to verify user sessions, making them susceptible to injection attacks. JWT is stateless, reducing the reliance on database queries after login.
2. **Embedded User Data**: User information is securely encoded in the token. The server validates user roles or permissions from the token instead of querying the database.
3. **Tamper-Proof Tokens**: The cryptographic signature of JWT ensures that tokens cannot be modified by attackers to inject malicious SQL.

3.3. Limitations and Best Practices

While JWT reduces the attack surface for SQL injection, it is not a complete solution. The following practices enhance security:

- Use parameterized queries or prepared statements for all database operations.
- Validate and sanitize all user inputs, including data encoded in JWT.
- Protect the secret key used for signing tokens and enforce token expiration policies.

3.4. Conclusion

JWT provides a robust mechanism for secure and stateless authentication, significantly reducing the risk of SQL injection attacks. By minimizing the need for session-based queries and leveraging cryptographic integrity, JWT enhances the security of web applications when combined with secure coding practices.

4. Threat Classification

The threat classification section of this report discusses the methodology used to identify and categorize potential cybersecurity threats within the context of the retail sector. This section outlines the approach taken to train, test, and evaluate the machine learning models used for classifying network traffic and POS system logs as normal or indicative of a potential attack.

4.1. Data Collection and Preprocessing

The first step in threat classification is acquiring relevant data that represents both legitimate and malicious activities. The dataset used in this project includes anonymized network logs and POS transaction data. These datasets contain multiple attributes that can be indicative of network or system behavior.Here we have used the **UNSW-NB15** dataset

Preprocessing involves several key steps:

- **Data Cleaning**: Removal of irrelevant or redundant columns and handling of missing values.
- **Feature Selection**: Identifying and selecting the most meaningful features for training the model. This is essential for improving model performance and reducing computational overhead.
- **Normalization**: Standardizing the scale of numerical features to ensure uniformity and facilitate model convergence.
- **Encoding**: Transforming categorical data into numerical formats using label encoding or one-hot encoding.

4.2. Model Architecture

The project implements a deep learning model that is capable of learning complex patterns in the data. The architecture of the model is designed to handle multi-dimensional input and learn from large datasets efficiently. Key design decisions include:

- **Input Layer**: Sized according to the number of features in the dataset.
- **Hidden Layers**: Comprising dense layers with activation functions such as ReLU to introduce non-linearity, and dropout layers to prevent overfitting.
- **Output Layer**: A softmax activation function to produce probability distributions across different threat classes (e.g., normal, attack type 1, attack type 2).

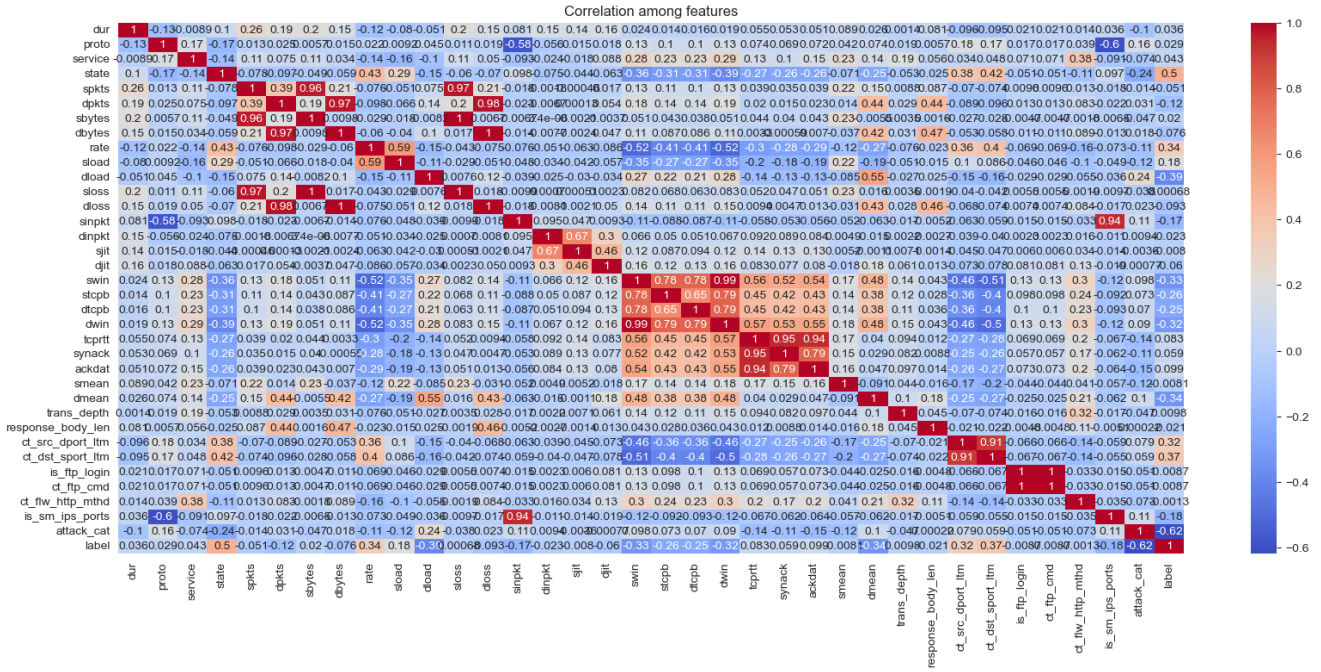

Figure 1. Correlation among the features of the dataset.

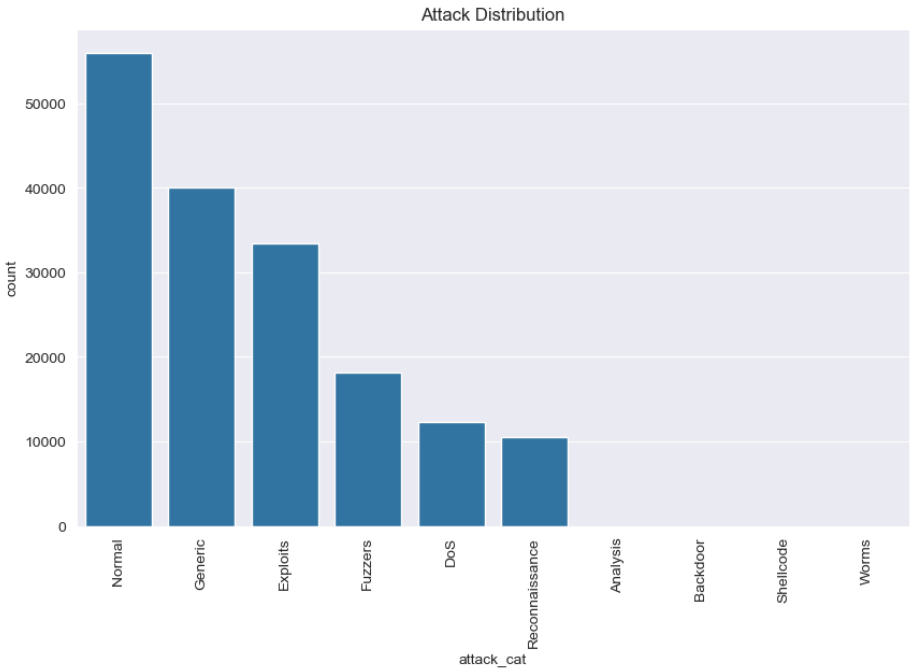

Figure 2. Attack distribution

4.3. Training and Evaluation

The training process involves using a labeled dataset to teach the model to distinguish between normal and malicious behavior. The model is trained using an appropriate optimization algorithm (e.g., Adam) with categorical cross-entropy as the loss function. During training:

- **Validation**: A validation split is used to monitor the model's performance on unseen data and avoid overfitting.
- **Hyperparameter Tuning**: Techniques such as grid search or random search are employed to find the best set of hyperparameters (e.g., learning rate, batch size).

  Evaluation metrics include:
- **Accuracy**: Percentage of correct predictions out of total predictions made.
- **Confusion Matrix**: Provides insights into true positives, false positives, true negatives, and false negatives.

4.4. Results and Performance Analysis

The results section presents a detailed analysis of how the trained model performed on the test data. This includes metrics such as testing accuracy and a visual representation of the confusion matrix. Performance analysis also includes a comparison between various model architectures, if applicable, to highlight the most effective configuration for this task.

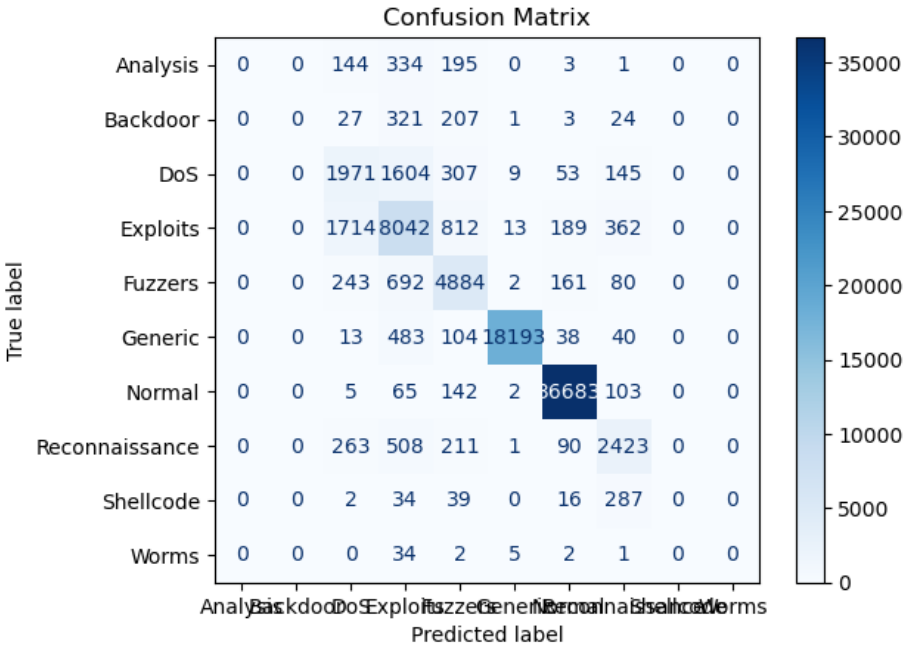

Figure 3. Correlation among the attacks after training

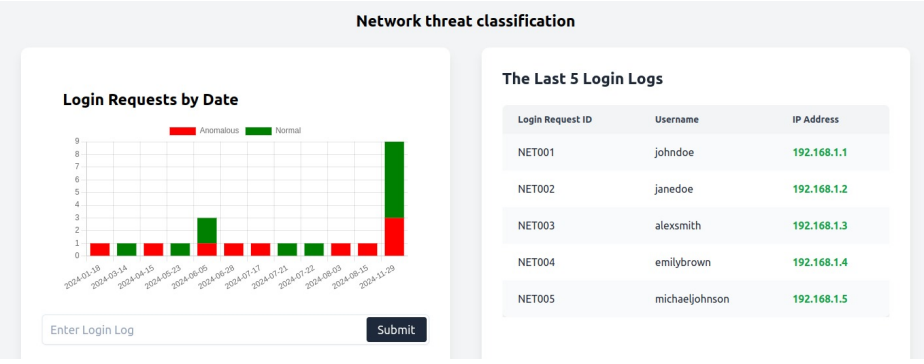

Figure 4. Dashboard with real time updates

5. Anomaly Detection

Anomaly detection is a technique used to identify patterns in data that do not conform to expected behavior. It is commonly used in various fields, including fraud detection, network security, and system monitoring. The core idea is to model normal behavior and then flag instances that deviate significantly from this baseline. These deviations, or outliers, are considered anomalies and often represent abnormal events or potential threats.

Anomaly detection in Point-of-Sale (POS) logs focuses on identifying unusual patterns in transaction data that could indicate fraudulent activities. By analyzing features like transaction amounts, timestamps, and payment methods, anomaly detection models, such as **Isolation Forest** or **Autoencoders**, can distinguish between normal and suspicious behavior. Fraudulent activities are often marked by irregularities, such as unusually large transactions or activities occurring at odd times. Techniques like **unsupervised learning** are particularly effective in detecting such outliers, as they do not require labeled data.

5.1. Dataset

To simulate real-world retail scenarios, we generated synthetic POS transaction logs with 1% labeled as fraudulent, incorporating features like transaction amounts, timestamps, payment methods, and item codes. These logs include both normal and suspicious data, with fraudulent transactions characterized by anomalies such as unusual amounts, irregular timings, or atypical order patterns. A Python script was used to generate valid and invalid timestamps, normal and suspicious transaction durations, and both common and rare item codes, allowing the simulation of various fraud scenarios like card testing, multiple small transactions, and data errors. By introducing anomalies with a higher probability and ensuring realistic variability, these synthetic logs provide a controlled environment for testing fraud detection techniques. The logs are saved in a CSV file for further use in training and evaluating fraud detection models.

5.2. Classification Model

In our project, we used the **Isolation Forest** model to detect anomalies in synthetic POS transaction logs, simulating fraudulent transactions. The goal was to identify outliers—transactions that deviate significantly from typical behavior, such as unusually high transaction amounts or suspicious purchasing patterns. We preprocessed the dataset by converting the timestamp to the transaction hour, selected relevant features, and removed non-numeric columns like `transaction_id`, `user_id`, and `payment_method`. For the categorical `item_code`, we applied `LabelEncoder` to convert it into numeric values, making the dataset compatible with the model.

We trained the **Isolation Forest** model with key parameters: `n_estimators` set to 100 (number of trees), `contamination` of 0.15 (indicating 15% of data is expected to be fraudulent), `max_samples` of 256 (samples per tree), and `bootstrap` set to True (data is sampled with replacement). `random_state` was fixed at 42 for reproducibility. After training, the model assigned an anomaly score to each transaction, with lower scores indicating anomalies. The model then classified each transaction as either normal (1) or fraudulent (0), based on its predictions.

5.3. Testing and Accuracy

The model's performance was evaluated by counting fraudulent versus normal predictions, which allowed us to isolate outliers that might indicate fraud or data errors. The **Isolation Forest** algorithm is particularly effective for detecting anomalies in high-dimensional, imbalanced datasets like POS logs, where fraudulent transactions are rare but critical to detect. This approach proved to be an essential tool in automating fraud detection systems.

Predicted\Actual	**Normal (0)**	**Fraudulent (1)**
Normal (0)	44811	189
Fraudulent (1)	2626	2374

Table 1. Confusion Matrix for the Isolation Forest model's predictions.

5.4. Conclusions

The Isolation Forest model demonstrated its effectiveness in detecting anomalies in synthetic POS transaction logs, simulating fraudulent transactions. However, the performance analysis, particularly through the confusion matrix, offers important insights into both the strengths and weaknesses of the model in fraud detection.

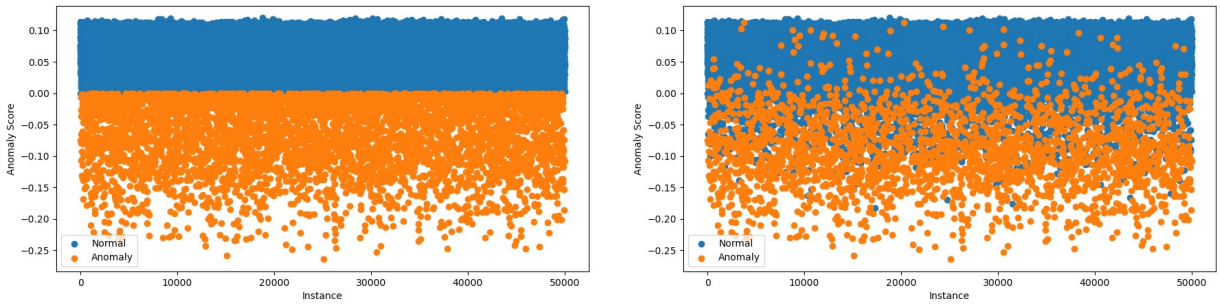

Figure 5. Left: Fraud detection using Isolation Forest. Right: Anomalies in the POS logs.

- **High Recall**: The model displayed high recall, detecting most fraudulent transactions, though it also flagged many normal transactions as fraudulent (high false positives). This makes it effective for identifying fraud, but at the cost of increased investigation efforts.
- **Low Precision**: The model resulted in a significant number of false positives, classifying normal transactions as fraudulent. This suggests that while it identifies fraud well, it also requires further tuning to reduce false alarms.
- **Effective for Imbalanced Data**: The Isolation Forest model is well-suited for handling imbalanced datasets like POS logs, where fraudulent transactions are rare. It efficiently detects outliers despite the imbalance in the data.
- **Room for Improvement**: Given the high recall and high false positives, adjustments like threshold tuning, additional features, or ensemble models could help improve the model's balance between recall and precision.
- **Real-World Applicability**: With further refinement, the model can be valuable for fraud detection systems in sectors like banking, e-commerce, and retail, where identifying fraudulent transactions quickly is crucial.

5.5. Demo

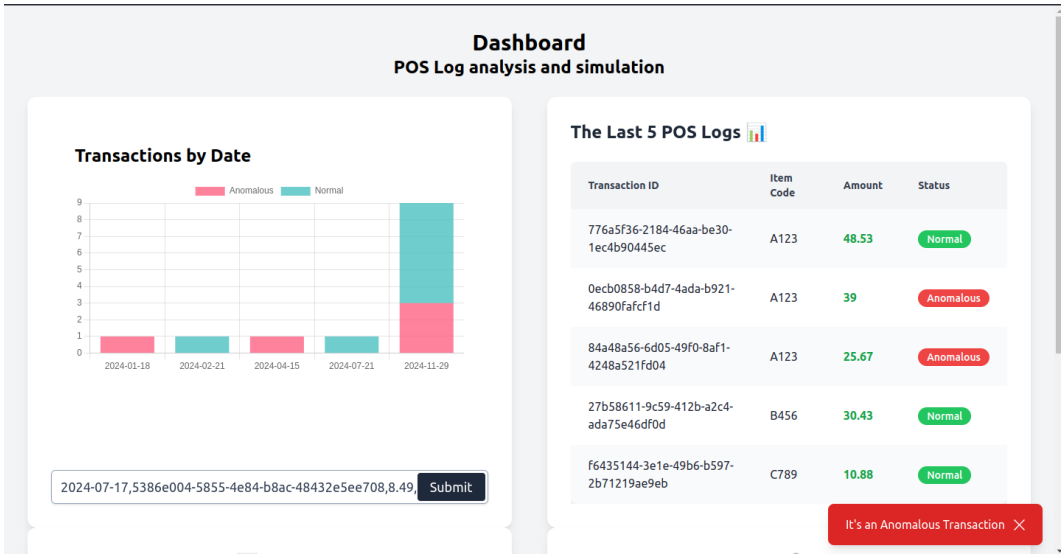

Figure 6. Dashboard with real time updates

Video Link: Demo Video - POS Logs Anomaly Detection.

6. Conclusions

The project on developing an AI-enhanced retail cybersecurity threat detection system concludes by demonstrating the effectiveness of advanced machine learning techniques, such as Isolation Forest and deep learning architectures, in safeguarding critical retail infrastructures. While the system achieves high recall in detecting fraudulent transactions and classifying

threats, its tendency toward false positives highlights areas for refinement. By integrating further optimization and advanced features, this approach has the potential to significantly enhance fraud detection accuracy and adapt to real-world scenarios. This study underscores the transformative role AI can play in mitigating cybersecurity risks in retail environments.