Antifraudster: A Multi-Perspective Real-Time Fraud Detection Framework for Multi-Participant E-Commerce Transactions

Yogesh N
Department of Information Science and Engineering
Acharya Institute of Technology
Banglore, India
yogeshn2005@gmail.com

Jagadish N
Department of Information Science and Engineering
Acharya Institute of Technology
Banglore, India
jagasimha.n@gmail.com

Sushma T M
Department of Information Science and Engineering
Acharya Institute of Technology
Banglore, India
sushmabilagi@gmail.com

Sahana R
Department of Information Science and Engineering
Acharya Institute of Technology
Banglore, India
sahanarrao2084@gmail.com

Spoorti Govind Nayak
Department of Information Science and Engineering
Acharya Institute of Technology
Banglore, India
nayakspoortig@gmail.com

Prashant C Bhosale
Department of Information Science and Engineering
Acharya Institute of Technology
Banglore, India
bhosaleprashant1703@gmail.com

Souparnika A S
Department of Information Science and Engineering
Acharya Institute of Technology
Banglore, India
souparnikasuraj6@gmail.com

Abstract– Fraud detection in e-commerce has been widely studied, and there are several models already that look at buyers, sellers, and transactions from different angles. These models, however have some key problems: they don't provide real-time monitoring, they use fixed thresholds that don't change with business size, they often need a lot of computing power that small businesses can't handle. To address these issues, this paper introduces Antifraudster, a new framework that improves multi-perspective fraud detection with three main features:(i) real-time fraud detection which give instant alerts, (ii) adaptive thresholds that adjust the fraud sensitivity based on vendor size, and (iii) an explainable AI layer that provides a clear reason for fraud predictions. The framework is also designed to be lightweight and efficient which makes it work well for both big e-commerce platforms and also smaller businesses. Tests show that Antifraudster improves detection accuracy, reduces false alarms, and builds more trust among vendors compared to existing systems. The recall for fraud detection is a bit lower (~82%) than the overall system accuracy of 90% which indicates that a small number of frauds might not be recognized immediately. Since the precision is still high(~90%), there are not many false alarms for valid transactions. Recall will be improved in future research in future work using combination machine learning techniques, more transaction metadata, and complex feature engineering.

Keywords – E-commerce, Fraud Detection, Multi-Perspective Transactions, Adaptive Thresholds, Explainable AI(XAI), and Real-Time System.

I.INTRODUCTION

E-commerce has transformed various global trade by enabling seamless digital interactions among buyers, sellers, and intermediaries. With billions of transactions taking place everyday on sites such as Amazon, Flipkart, and Shopsy, fraud has emerged as one of the biggest threats to online marketplace sustainability and trust. Fake return claims, fictitious product listing, buyer-seller collusion, account abuse, and transaction record manipulation are examples of some common frauds. According to the reports, e-commerce fraud already costs the global economy billions of dollars every year.

Researchers and industry have developed fraud detection systems that use machine learning models, rule-based approaches, and anomaly detection to reduce such threats. Recently, frameworks for multi-perspective fraud detection have been introduced.

A. Problem

Multi-perspective models are a big step forward, but they still have issues:

They can't spot fraud as it happens; they usually need to be run later on a group of data. And they use the same rules for everyone, which doesn't work since bigger and smaller sellers have very distinct risk profiles. Making them unsuitable for vendors of different scales. They need a lot of computing power, which small sellers often can't get.And also they are hard to understand, like a mystery box.

Because of these many things, fraud still goes unnoticed, or sellers stop trusting the systems because of too many mistakes.

B. Motivation

There is a huge need for a fraud detection system that works for all sizes of businesses.Big e-commerce sites need monitoring systems that can grow, vendors need tools that are light, use few resources, and can work on a limited infrastructure. Also, companies want AI that can explain why it thinks something is fraud so they can trust it more. That's why we're making a system that can find fraud in real-time, make changes to different situations, and explain its decisions from different points of view.

C. Contributions

This paper presents Antifraudster, a multi-perspective fraud detection framework with the following advantages:

A multi-perspective fraud detection engine that incorporates contextual, transaction, buyers, and seller data.

A pipeline for real-time monitoring that can produce fraud alerts. And an adaptive thresholding system that adjusts fraud sensitivity across vendors.

A layer of Explainable AI(XAI) that offers explanations for fraud predictions that are readable by humans. An implementation that is resource-efficient and appropriate for both large platforms and small vendors.

The rest of this paper is organised as follows:

The suggested Antifraudster framework is described in detail in Section III following a review of related tasks in Section II. Section IV discusses the analytical techniques used to detect fraud. Section V presents the model analysis, and Section VI discusses the specifics of implementation. Results and discussion are presented in Section VIII after case studies and scenarios are illustrated in Section VII. Section IX concludes by discussing the conclusion and future work.

II. RELATED TASKS

A variety of methods, including rule-based systems, sophisticated machine learning, and multi-perspective models, have been used to detect the fraud in e-commerce. Even though each helps to enhance fraud detection, there are still significant drawbacks.

A. Rules Systems Based

The first fraud detection system was rule-based and relied on predetermined cutoff points like transaction limits, frequency of purchases, or geographic limitations. These systems are rigid but computationally cheap. By making minor behavioural changes, fraudsters can get around them, increasing the number of false positives and false negatives.

B. Machine learning and anomaly detection

Large datasets made it possible to employ machine learning techniques like ensemble classifiers, decision trees, logistic regression, and support vector machines(SVM). Compared to static rules, these models are better at capturing nonlinear transaction behaviours. In a similar vein , anomaly detection methods find odd variations in transaction patterns. These approaches, however, are usually offline, single-perspective(focusing on buyer-only or seller-only data), and act as black boxes with little interpretability.

C. Multi-Perspective Models

More recently, research has looked into multi-perspective fraud detection, which combines data from the buyers, sellers, and intermediaries to identify the complex fraud scenarios. Process mining and also graph-based models have been used to analyse transaction flows and identify irregularities among participants. Although these methods increase the coverage of fraud detection,they have three major disadvantages: they cost a lot of money, they typically don't work in real time, and they give end users little to no explainability. As a result, their use in the small and medium-sized businesses is still limited.

D. Research Gap

The gaps listed below are evident from the literature review;

Real-Time Detection: Current models identify fraud retrospectively as opposed to immediately.

Adaptive Threshold: Static thresholds used in current systems are unable to adjust to vendors of varying size.

Explainability: The majority of fraud detection techniques are still opaque, which reduces user confidence and usability.

Resource Efficiency: Not many systems are made to be portable and accessible to small vendors without substantial infrastructure.

The creation of Antifraudster, a novel multi-perspective framework that incorporates explainable AI, adaptive thresholds, and real-time monitoring, while remaining resource-efficient.

III. PROPOSED FRAMEWORK: ANTIFRAUDSTER

Our proposed framework: Antifraudster, improves on the previously discussed systems by offering the implementation of multi-perspective analysis (among users and transactions) with continuous analysis and monitoring, adaptive thresholds, and explainable AI. Antifraudster will encode patterns of fraud against fraudsters, buyers, sellers, transactions etc., in a scalable framework that will be acceptable to large platforms and small vendors alike.

A. System Model

Antifraudster will be applied in a multi-participant, e-commerce transaction environment with buyers, sellers, payment providers, and logistics intermediaries. A single e-commerce transaction will produce various types of data including financial logs, user logs of activities, time stamps, geographic data, and contextual metadata (e.g. IP addresses, and processing device types). Fraud, depending on the context could occur from any one participant, or the collusion of multiple participants. Antifraudster provides an opportunity to examine transactional behavior from a composite perspective.

B. System Architecture

Antifraudster has the following layers in the context of the system architecture:

- **Data Collection Layer:** collects transaction logs, user activity logs, and contextual metadata logs via API calls, or an engagement with the platform.
- **Preprocessing Module:** cleaning, normalization, and standardization of different data sources.
- **Feature Extraction Layer:** Extracts features, which consist of identifying abnormal pricing, identifying unusual frequency of returns, identifying failed payments, mismatched IP, or identifying sell fulfilment ratios.
- **Fraud Engine:** Applies multiple machine-learning classifiers (SVM, Decision Tree, Logistic Regressions, Na¨ıve Bayes) with ensemble learning.
- **Adaptive Thresholding Module:** Adapts fraud scoring cutoffs by vendor size and transaction volume.
- **Explainable AI (XAI) Layer:** Generates interpretable explainers for flagged cases to create trust and inform decision making.
- **Real-Time Monitoring and Alerts:** Analyzes live streams of transactions and shows up on vendor dashboards with fraud alerts within fractions of second latency, and tagged further for latency and performance monitoring.

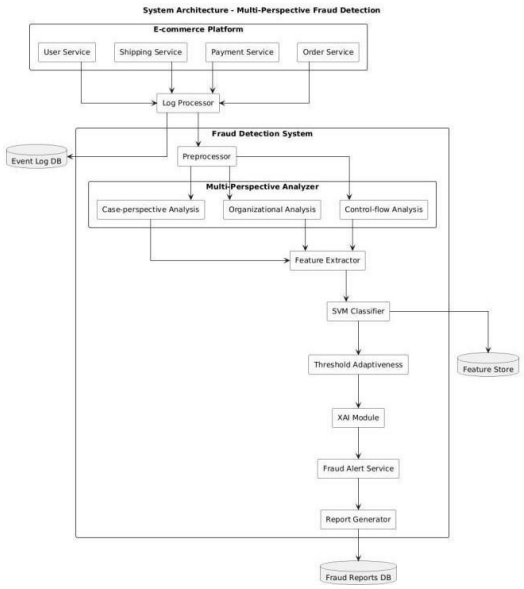

Fig. 1. System Architecture

C. Data Preprocessing

Raw transaction data often contains noise, missing values, or inconsistent formats. The preprocessing step involves:

- Removing duplicate or incomplete records.
- Handling missing values using imputation.
- Encoding categorical variables such as device type and payment method.
- Normalizing numerical attributes such as transaction amount and frequency.

D. Feature Engineering

Antifraudster develops multi-perspective attributes for analysis:

- Seller Features: fulfillment rate, odd pricing behaviours, complaints ratio.
- Transaction Features: purchase time of transaction, price offsets from market averages, confirmation delays.
- Network Features: dense links between sellers and buyers, indicating collusion.

E. Fraud Detection Engine

Antifraudster's fraud detection engine combines classifiers to provide a balance in interpretation, accuracy, and execution:

- Support Vector Machine (SVM): can capture more complex decision boundaries of subtle fraud patterns.
- Decision Tree: interpretable rules regarding fraud detection.
- Logistic Regression: estimates the probabilities for fraud vs. legitimate transactions.
- Naive Bayes: a fast, probabilistic model, applicable to real-time or fast-streamed data that is sizable.

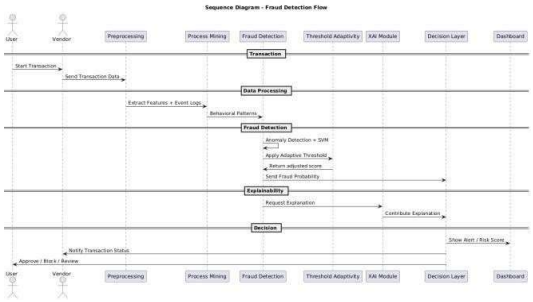

Fig. 2. Interaction flow of the transaction process.

F. Adaptive Thresholding

Normal models based off of fixed thresholds (example, fraud score > 0.7), where Antifraudster is adaptive in thresholds:

- Lower thresholds for smaller vendors, with fewer transactions to capture more subtle anomalous transactions.
- Higher thresholds for larger vendors to deal with more transactions to reduce false positives.

G. Explainable AI (XAI)

The XAI layer enhances transparency by generating short explanations for fraud alerts. Examples include:

- "Transaction flagged: 3 failed payment attempts and suspicious IP mismatch."
- "Seller flagged: 92% non-fulfillment rate and abnormal pricing patterns."

H. Real-Time Monitoring

Antifraudster processes incoming transactions in real time, with sub-second latency. Fraudulent activity is flagged immediately, and alerts are displayed on the vendor dashboard. This prevents fraudsters from exploiting time delays that exist in offline or batch detection systems.

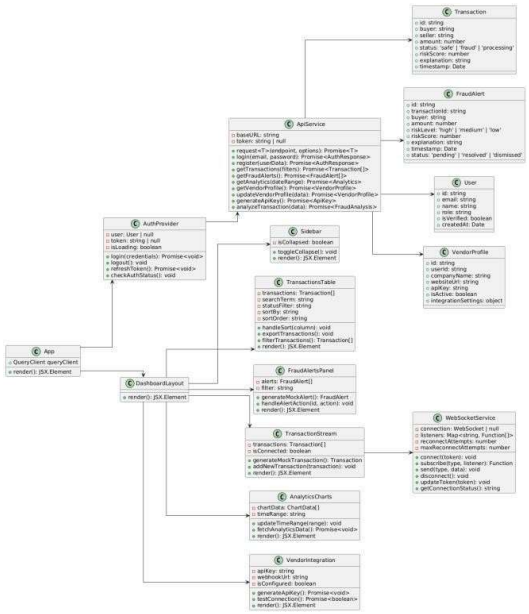

Fig. 3. Class diagram

IV. ANALYTICAL METHODS

To detect fraud in real time, the suggested framework Antifraudster combines machine learning classification, Petri net modelling, anomaly extraction, and multiperspective process analysis. Antifraudster's analytical pipeline guarantees that every transaction is dynamically assessed prior to payment completion, in contrast to conventional offline systems.

A. Theoretical Basis

Every transaction by a trace in an event log, which is determined by the following tuple:

δ = {TransactionID, Timestamp, Action, Actor, Resource, Context}.

Traces record all aspects of a transaction, such as platform notifications, payment events, and seller confirmations. The transaction workflow is represented by a labelled Petri net.

Definition 1 (System Net):

$$SN = (IP\ N, Minit, Mf\ inal)$$

Where IP N = (P, T, F, I) is a Petri net with transactions T, places P, and flow relation F is the definition of a system net. Transitions to the universe of actions UA are mapped by the labelling function $l: T \rightarrow UA$. Initial and final markings are represented by *Minit and Mfinal.*

Definition 2(Date Petri Net): A Data Petri Net(DPN) is an extension of SN that includes guard conditions G, read/write functions R and W, and data variables V.

$$DP\ N = (SN, V, U, R, W, G)$$

Guard conditions determine each transition $t \in T$, guaranteeing that fraudulent deviations(such as unusual pricing or missing fulfillment) are recorded.

B. Conformance Checking

Verification of Conformance Event logs are compared to the Petri net model of valid transactions in order to identify anomalies. Differences between the model and log are noted:

- Control Flow Deviation: The absence of a necessary transaction step, such as a shipping confirmation.
- Time Deviation: when the adaptive threshold is exceeded by the throughput time between order and payment.

B. Adaptive Thresholding Integration

Conventional models employ a static fraud score cutoff. The anomaly score in Antifraudster is dynamically modified:

$\theta_v = \alpha \cdot 1/\sqrt{N_v}$

C. Integration of Explainable AI(XAI)

The anomaly features are enhanced aith explanations to improve interpretability. For instance:

- "Transaction flagged: 45% abnormal price deviation from market average."
- "Seller flagged: repeated cancellations and fulfilment ratio below 70%."

D. Algorithm for Anomaly Extraction

Aggregation, and conformance deviations are all combined in the anomaly extraction process.

Algorithm 1: Antifraudster Multiperspective Anomaly Extraction-

1) Input: adaptive threshold θ_v, Data Petri Net DP N, and event log L.

2) On DP N, replay every trace $\delta \in$ L and note any deviations:

- $C[\delta]$ = 1 if control flow deviation.
- $T[\delta]$ = 1 if time deviation (beyond θ_v).
- $R[\delta]$ = 1 if resource mismatch.
- $D[\delta]$ = 1 if contextual data mismatch.

3) F[δ] = (C, T, R, D) is an aggregate anomaly feature.
4) Enter F[δ] into the ensemble machine learning (ML) fraud detection engine.
5) Output: Fraud label with XAI explanation (legitimate or fraudulent).

E. Fraud Detection via Ensemble Classifiers

Each anomaly feature vector F[δ] is evaluated using an ensemble of classifiers:

- Support Vector Machine (SVM): captures complex
- decision boundaries.
- Decision Tree (DT): provides interpretable rules.
- Logistic Regression (LR): estimates fraud probabilities.
- Na¨ıve Bayes (NB): offers fast predictions for real-time streaming.

The ensemble combines predictions as:

$$Score(\delta) = \frac{1}{n}\sum_{i=1}^{n} h_i(F[\delta])$$

Where h_i are base classifiers. If $Score(\delta) > \theta_v$, the transaction is flagged.

V. MODEL ANALYSIS

Multiperspective analysis, adaptive thresholds, explainable reasoning, and real-time decision-making are all integrated into the suggested Antifraudster framework. This section examines the model's efficacy, behaviour, and design from a variety of angles.

A. Architecture Validation

The framework is validated against existing fraud detection approaches, which are often limited to static rules or batch based anomaly detection. Antifraudster introduces a layered design:

- Data Layer: collects transaction event logs (time, location,amount, resource usage).
- Processing Layer: applies conformance checking and anomaly extraction.
- Decision Layer: classifies fraud vs. safe transactions using an SVM-based model with adaptive thresholds.
- Explainability Layer: provides human-readable fraud reasons to small-scale vendors.

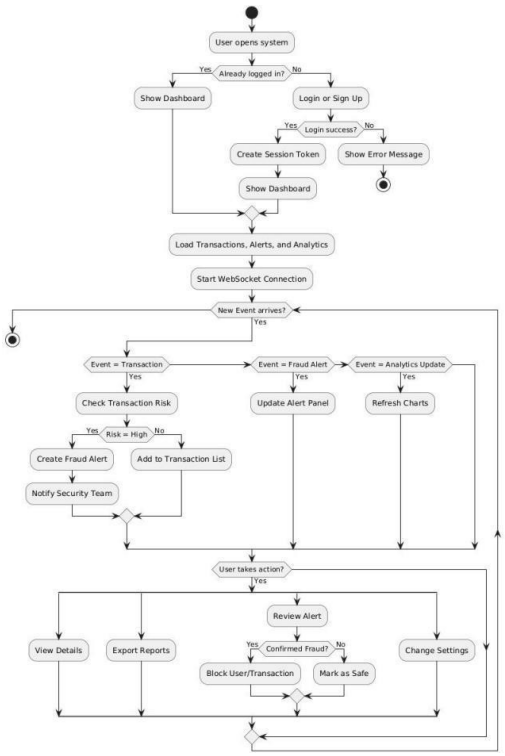

Fig. 4.Workflow Diagram of the Anti-Fraud System

B. Feature Contribution Analysis

Feature importance analysis was used to assess the contribution of various viewpoints. The most significant characteristics were transaction time, suspicious order modifications, mismatched amounts, and unusual IP changes. By dynamically modifying fraud cutoffs according to vendor transaction volume, adaptive thresholding improves detection even more.

C. Analysis of Complexity

The model is made to function in settings with limited resources. While SVM classification yields results in less than a second, event log preprocessing and anomaly extraction take O(n) time in relation to the number of transactions.

D. Advantages and Disadvantages

Among the framework's advantages are:

- Real-time detection prior to payment confirmation.
- Explainability (XAI) to increase vendor confidence.
- Flexibility for online shopfronts of modest size.

This layered approach ensures both efficiency and inter pretability.

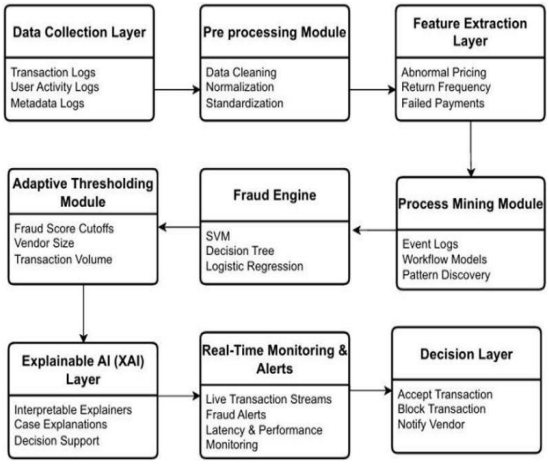

Fig. 5. Module Design

Among the drawbacks are the requirement for regular retraining of the SVM model as fraud patterns change and the reliance on precise event log data. These can be addressed in future research by combining blockchain technology with online learning models to create tamper- proof event logs.

A. Technology Stack

The system was built with the following technology stack:

• Backend: The technology we used is Django, a Python web framework, for all API and the integration of the machine learning models with the web application.

• **Database:** MySQL for structured information of transactions, buyer/seller logs, fraud alerts, etc.

• **Frontend:** React with Bootstrap for the vendor dashboard so it could be responsive, and show the fraud alerts and explanations.

• **Machine Learning Libraries:** Scikit-learn for classification algorithms; TensorFlow/Keras for prototyping; Pandas and NumPy for data and data manipulation.

• **Environment:** Used Jupyter Notebook for the experimentation and venv (Python virtual environment) for package management.

B. Dataset

The dataset incorporates several publicly available e-commerce fraud datasets and synthetic data created to emulate multiparticipant transactions. Each transaction record included:

Transaction ID, Buyer ID, Seller ID.

Transaction amount, product category, and timestamp.

Buyer attributes: payment attempts, returns, IP address.

Seller attributes: fulfillment ratio, complaints, non-standard pricing.

Contextual attributes: time of day, device type, geographic mismatch.

This allows for fraud detection to be assessed holistically as it considers buyers, sellers and transactions.

C. Training and Testing Split

The dataset was split into:

Training set: 70% of dataset. Used to train the machine learning models.

Testing set: 30% of dataset. Used to test the performance on unseen transactions.

This split allows for a good balance of learning from the data and generalizing from it.

D. Algorithms

The fraud detection engine will use a variety of classifiers:

• **Support Vector Machine (SVM):** Great at finding non-linear decision boundaries.

• **Decision Tree Classifier:** Provides decision rules that are interpretable by humans.

• **Logistic Regression:** Appropriate for a binary fraud vs abuse classification.

• **Naïve Bayes:** Very fast, scalable, and is well suited for a real-time data streaming context.

E. Algorithm Use

The transaction fraud detection lifecycle is as follows:

1. Transaction data collected and merged.
2. Features engineered and served as input to the classifiers.
3. Each classifier generates a fraud probability score.
4. The ensemble model takes the fraud probability scores from each classifier.
5. Adaptive thresholding is used to modify/determine the decision band depending on vendor size.
6. The XAI module provides an explanation and posts to the dashboard.

VI. CASE STUDIES AND SCENARIOS

A. Onboarding and Vendor Integration

Using the signup portal, a small-scale clothing vendor registers on Antifraudster. The vendor links their checkout system to Antifraudster using the supplied API key. Prior to payment confirmation, every incoming transaction is subsequently automatically passed through the fraud detection engine.

B. Real-Time Transaction Monitoring
A seller starts a transaction in which several purchasers pay at the same time. Live transaction entries are displayed in the dashboard's real-time Transaction stream.

Antifraudster uses its fraud detection pipeline prior to payment confirmation:

- Green-highlighted transactions are safe and can move forward.
- Transactions deemed suspicious (such as several orders placed by the same IP address in a single second) are marked in yellow and comes with an explanation.
- With warnings like "IP mismatch with user account" or "Unusual purchase amount beyond adaptive threshold," fraudulent transactions are blocked(red).

C. Fraud Blocking Based on Thresholds
A buyer tries to buy INR 50,000 worth of goods from a seller whose average transaction value is less than INR 5,000.

D. Explainable AI(XAI) Scenario
In another instance, a frequent customer unexpectedly logs in from a foreign nation and makes several attempts to modify the shipping address prior to checking out. After identifying these irregularities, the system flags the transaction as suspicious . The XAI explanation panel display:

1) IP does not correspond to historical records.
2) Tampering of the address was discovered.
3) Transaction timing that deviates from historical patterns.

VII. RESULTS AND DISCUSSION

With the help of synthetic multi-participant transaction data in conjunction with the benchmark e-commerce datasets, Antifraudster's performance was analyzed. The system was measured on the basis of classification accuracy, flexibility with respect to vendor sizes, decision transparency in fraud detection, and real-time responsivenes

A. Evaluation metrics
The following metrics were applied to measure efficacy:

- **Accuracy:** The proportion of correctly classified transactions.
- **Precision:** Ratio of flagged cases that were truly fraudulent.
- **Recall:** The proportion of real fraud cases that were successfully detected.

B. Comparative Analysis
Antifraudster yielded more results than current multi-perspective systems in the following ways:

- Higher accuracy (~90%)
- Adaptive thresholds decrease false positives.
- XAI offered transparent fraud alerts that were not available in the earlier systems.
- Real-time detection as opposed to methods that rely solely on the past.

VIII. CONCLUSION AND FUTURE WORK

Complex fraud scenarios involving buyers, sellers, and intermediaries continue to persistently affect e-commerce platforms. While current multi-perspective fraud detection systems offer some partial approaches, they often fall short in terms of flexibility, explainability, and real-time responsiveness.

A. Future Work
Despite Antifraudster's positive outcomes, there a still a number of areas that require further research:

- **Blockchain Integration:** Including blockchain based transaction validation for greater traceability and trust.
- **Advanced Deep Learning Models:** Looking into transformers and graph neural networks(GNNs) for demonstrating relationships among participants in vast marketplaces.
- **Cross-Platform Fraud Detection:** Expanding the framework to recognise the fraudulent activity that is observed across several e-commerce platforms.
- **Explainability advancements:** Introducing the ability to create human-friendly narratives and visualizations for fraud alerts with the help of the XAI module.
- **Deployment at scale:** Validating scalability and robustness through practical evaluations of active e-commerce systems.

ACKNOWLEDGEMENT

We want to thank our guide, Prof. Yogesh N, for always being there for us and to help us finish this work. We are also grateful for the help and resources from the Department of Information Science and Engineering. Finally, thanks to our friends and classmates for their useful advice as we worked on this project.

REFERENCES

[1] Y. Zhao, Z. Zheng, J. Wu, J. Guo and X. Liu, "A Multi- perspective Fraud Detection Method for Multi-participant E-commerce Transactions," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 1, pp. 305–317,
Jan. 2023.
[2] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review," Statistical Science, vol. 17, no. 3, pp. 235–255, 2020.
[3] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning,* vol. 20, no. 3, pp. 273–297, 1995.
[4] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986. [5] D. W. Hosmer and S. Lemeshow, "Applied Logistic Regression," *John Wiley & Sons*, 2nd ed., 1997.
[6] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning*, vol. 29, no. 2, pp. 103–130, 2004.
[7] M. Ahmed, A. N. Mahmood and J. Hu, "A Survey of Network Anomaly Detection Techniques," *Journal of Network and Computer Applications,* vol. 60, pp. 19–31, Jan. 2017.
[8] W. M. P. van der Aalst, "Process Mining: Discovery, Conformance and Enhancement of Business Processes," *Springer,* 2011.
[9] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," arXiv preprint *arXiv:1702.08608,* 2019.
[10] S. M. Lundberg and S. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems (NeurIPS),* pp. 4765–4774, 2017.
[11] H. V. Jagadish et al., "Big Data and Its Technical Challenges," *Communications of the ACM,* vol. 57, no. 7, pp. 86–94, 2014.
[12] S. Bhattacharyya, S. Jha, K. Tharakunnel and J. C. Westland, "Data Mining for Credit Card Fraud: A Comparative Study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, Feb. 2011.