A PROJECT REPORT ON

A MULTI-VIEW FRAUD DETECTION APPROACH FOR MULTI-PARTICIPANT E-COMMERCE TRANSACTIONS

In partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

In

CSE (ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)

SUBMITTED BY

P. CHENNARAO	**(21JD1A4542)**
K. RAMESH	**(21JD1A4525)**
D. PRASANNA KUMAR	**(21JD1A4513)**
SK. RIZWANA	**(21JD1A4551)**

Under the esteemed guidance of

K .Vaddi Kasulu M .TECH(CSE), Ph .D(CSE)

Head of the Department

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & DATA SCIENCE)

ELURU COLLEGE OF ENGINEERING AND TECHNOLOGY

DUGGIRALA (V), PEDAVEGI (M), ELURU-534004

APPROVED BY AICTE-NEW DELHI & AFFLIATED TO JNTU-KAKINADA

2021 - 2025

ELURU COLLEGE OF ENGINEERING & TECHNOLOGY

(Affiliated to JNTU-KAKINADA, Approved by AICTE-NEW DELHI)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)

CERTIFICATE

This is to certify that the Project Report entitled **"A Multi-View Fraud Detection Approach for Multi-Participant E-commerce Transactions"** being submitted in partial fulfilment for the award of the degree of **Bachelor of technology** in **CSE (ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)** to the **Jawaharlal Nehru Technological University, Kakinada** is a record of bonafide work carried out **by P.Chennarao(21JD1A4542), K.Ramesh(21JD1A4525), D.Prasanna Kumar (21JD1A4513), SK.Rizwana (21JD1A4551)** under the guidance and supervision .

PROJECT GUIDE **HEAD OF THE DEPARTMENT**

K .Vaddi Kasulu MTECH (CSE), Ph .D

Professor & HOD

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The present project work is the several days study of the various aspects of the project development . During this the effort in the present study, we have received a great amount of help from our **Chairman Sri V .RAGHAVENDRA RAO** and **Secretary V .RAMA KRISHNA RAO** of **Padmavathi group of Institutions,** which we wish to acknowledge and thank from depth of our hearts .

We are thankful to our **Principal Dr . P . BALAKRISHNA PRASAD** for permitting and encouraging us in doing this project .

We are deeply intended to **Sri Dr . K . VADDI KASULU** M Tech ., Ph .D . **, Professor & Head of the Department**, whose motivation and constant encouragement has led to pursue a project in the field of software development .

We are very much obliged and thankful to our project guide **K . VADDI KASULU** M .Tech ., Ph .D . MTECH(CSE), **(Professor & HOD),** for providing this opportunity and constant encouragement given by her during the course . We are grateful to her valuable guidance and suggestions during our project work .

Our parents have put our self ahead of themselves . Because of their hard work and dedication, we had opportunity beyond our wildest dreams . Our heartfelt thanks to them for giving us all we ever needed to be successful student and individual .

Finally, we express our thanks to all our other faculty members, classmates, friends and neighbors who helped us for the completion of our project and without infinite love and patience this would never have been possible .

P. CHENNARAO **(21JD1A4542)**
K. RAMESH **(21JD1A4525)**
D. PRASANNA KUMAR **(21JD1A4513)**
SK. RIZWANA **(21JD1A4551)**

ABSTRACT

Detection and prevention of fraudulent transactions in e-commerce platforms have always been the focus of transaction security systems . However, due to the concealment of e-commerce, it is not easy to capture attackers solely based on the historic order information . Many researches try to develop technologies to prevent the frauds, which have not considered the dynamic behaviors of users from multiple perspectives . This leads to an inefficient detection of fraudulent behaviors . To this end, this paper proposes a novel fraud detection method that integrates machine-learning and process mining models to monitor real-time user behaviors . First, we establish a process model concerning the B2C e-commerce platform, by incorporating the detection of user behaviors . Second, a method for analyzing abnormalities that can extract important features from event logs is presented . Then, we feed the extracted features to a Support Vector Machine (SVM) based classification model that can detect fraud behaviors . We demonstrate the effectiveness of our method in capturing dynamic fraudulent behaviors in e-commerce systems through the experiments .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

UML – UNIFIED MODELING LANGUAGE

DT – DECISION TREE

GUI – GRAPHICAL USER INTERFACE

SVM – SUPPORT VECTOR MACHINE

1.INTRODUCTION

1.1 INTRODUCTION

E-commerce platforms have revolutionized the way businesses and consumers interact, offering seamless transactions and global accessibility. However, as online transactions continue to rise, so do fraudulent activities. Cybercriminals exploit vulnerabilities in e-commerce systems, leading to financial losses, reputational damage, and a decline in customer trust . Traditional fraud detection techniques primarily rely on analyzing historical order data, which often fails to capture evolving fraudulent patterns. To address this challenge, this project introduces **" A Multi-View Fraud Detection Approach for Multi-Participant E-commerce Transactions."** Unlike conventional fraud detection methods that focus solely on transaction history, our approach integrates **machine learning** and **process mining models** to analyze real-time user behaviors . By leveraging a **Support Vector Machine (SVM)-based classification model**, we enhance the accuracy of fraud detection, ensuring proactive identification of suspicious activities . This research aims to establish a **process model** for Business-to-Consumer (**B2C**) e-commerce platforms, incorporating user behavior analysis and event log extraction techniques . By combining multiple perspectives of user interactions, our system improves fraud detection efficiency and reduces false positives . The effectiveness of our approach is demonstrated through experimental evaluations, showcasing its potential in strengthening e-commerce transaction security .

1.2 PURPOSE OF THE PROJECT

The purpose of the project **" A Multi-View Fraud Detection Approach for Multi-Participant E-commerce Transactions "** is to develop an advanced fraud detection system that can accurately identify fraudulent transactions by analyzing real-time user behavior s . Traditional fraud detection methods primarily rely on historical transaction data, which often fails to capture **dynamic fraud patterns** . By leveraging **machine learning algorithms** and **process mining techniques**, this project aims to enhance the accuracy and efficiency of fraud detection in e-commerce platforms . The proposed system integrates **Support Vector Machine (SVM)** for fraud classification, utilizing extracted behavioral features from event logs . This approach enables the detection of fraudulent activities based on both historical trends and real-time anomalies . The primary objective is to equip **e-commerce platforms, financial institutions, and security teams** with a robust fraud detection mechanism that can **reduce financial losses and improve transaction security**

1.3 EXISTING SYSTEM

In Existing Systems, machine-learning-based methods learn from historical data to classify or predict future observations, identifying potentially risky offline or online transactions . Xuetong Niu et al . conducted a comparative study on credit card fraud detection methods that rely on machine-learning algorithms . Most machine-learning models perform well on datasets of credit card transactions . Moreover, supervised models perform slightly better than unsupervised models after additional pre-processing, such as removing outliers .

Credit card fraud detection is widely deployed at the application layer, using the idea of detecting specific abnormal user behaviours to identify fraudulent activities . Supervised learning algorithms are the most commonly used methods for online fraud monitoring transactions due to their higher accuracy and broader coverage. Recent research has demonstrated that machine learning methods efficiently capture fraudulent transactions in credit card applications

Fraudsters frequently alter their behavioural patterns dynamically to bypass existing fraud detection methods . In online credit card fraud detection, SVM can classify user behaviours under complex scenarios and deliver reliable results . Many researchers leverage multiple detection methods to enhance fraud detection . For example, focusing on payment fraud applications, Dahee Choi et al. proposed a method that combines supervised and unsupervised learning . Most machine learning-based methods use historical data to analyse fraudulent transactions but do not emphasize transactional process flow and dynamic user behaviours.

Disadvantages: 1 . More likely prone to Under fitting and for Outliers .

2 . Less Accuracy .

1.4 PROPOSED SYSTEM

The proposed system combines the advantages of process mining and machine learning models by introducing a hybrid method to solve the anomaly detection in data flows, which provides information about each action embedded in a control flow model. By modeling and analyzing the business process of the e-commerce system, this method can dynamically detect changes in user behaviors, transaction processes, and noncompliance situations, and comprehensively analyze and identify fraudulent transactions from multiple perspectives. Important contributions of this paper are listed as follows:

1) A conformance checking method based on process mining is applied in the field of e-commerce transactions to capture the abnormalities.

2) A user behavior detection method is proposed to perform comprehensive anomaly detection based on Petri nets.

3) An SVM model is developed by embedding a multi perspective process mining into machine learning methods to automatically classify fraudulent behaviors.

The **Decision Tree Regressor** is used to model intricate relationships in fraud detection data, allowing for a more flexible approach compared to linear models . Fraudulent activities often exhibit **non-linear patterns**, and Decision Trees can naturally detect these anomalies without assuming predefined data distributions . Additionally, Decision Tree models are **less sensitive to outliers**, ensuring more **accurate fraud predictions** even when unusual transaction patterns occur .

The **Decision Tree Classifier** is utilized to **handle imbalanced datasets**, where fraudulent transactions are significantly fewer than legitimate ones . Unlike traditional models that require additional balancing techniques, Decision Trees can **partition data efficiently** to create meaningful decision boundaries . Moreover, the classifier can incorporate **domain knowledge** to enhance detection accuracy by integrating insights from transaction history and fraud detection experts .

Advantages:

Less prone to underfitting – Captures complex fraud behaviours more effectively
Handles complex data – Adapts to intricate and dynamic fraud patterns .
Higher model accuracy – Provides precise fraud detection and classification .

Problem Statement : E-commerce platforms face fraud from multiple participants, like buyers, sellers, and payment processors. Traditional fraud detection methods miss complex patterns by focusing on limited data. This project aims to develop a multi-view approach that analyzes transactions, user behaviour, and networks to detect fraud more accurately

2.LITERATURE SURVEY

2.1 Application of Isolation Forest for Credit Card Fraud Detection:

The Isolation Forest algorithm has been effectively utilized to detect anomalies in datasets, such as identifying fraudulent credit card transactions. A study by Andrea Dal Pozzolo et al. (2015) applied this algorithm to a dataset comprising 284,807 transactions, of which only 492 were fraudulent (0 .172%). The study highlighted the challenges posed by such imbalanced data and emphasized the need for specialized metrics like the Area Under the Precision-Recall Curve (AUPRC) for accurate evaluation. The results underscored the algorithm's capability to isolate outliers, although challenges like low precision due to false positives were noted.

2.2 Ensemble Learning Techniques in Fraud Detection:

Ensemble learning, which combines multiple classifiers to improve model robustness, has been proposed as an efficient technique for detecting fraudulent activities in banking and credit card systems . Francisco Louzada and Anderson Ara (2012) introduced Bagging k-dependence probabilistic networks as a powerful fraud detection tool, demonstrating improved detection rates .Similarly, G . Ganesh Sundarkumar and Vadlamani Ravi (2015) proposed a novel hybrid undersampling method for mining unbalanced datasets in banking and insurance, addressing the challenges of data imbalance in fraud detection scenarios .

2.3 Predictive Modeling of Financial Distress Using Ensemble Classifiers:

Accurately predicting business failures is crucial in financial decision-making. Yoonseong Kim and So Young Sohn (2012) applied ensemble classifiers to analyze stock market data, effectively detecting suspicious symptoms of stock price manipulation. Their study demonstrated that ensemble methods could enhance the reliability of financial distress predictions, thereby aiding in proactive financial decision-making.

2.4 Hybrid Sampling and Ensemble Approaches for Imbalanced Datasets:

Addressing the challenge of imbalanced datasets in fraud detection, G. Ganesh Sundarkumar and Vadlamani Ravi (2015) introduced a hybrid undersampling method that combines ensemble learning with sampling techniques. This approach effectively improved the detection of fraudulent cases by balancing the dataset, thereby enhancing the model's performance

2.5 Enhancing Fraud Detection with Advanced Ensemble Methods:

Recent advancements in ensemble learning have led to the development of more sophisticated methods for fraud detection. These methods focus on improving the robustness of normal behavior modeling, thereby enhancing the detection of fraudulent activities in banking and credit card systems. The integration of ensemble classifiers has proven to be an effective strategy in identifying complex fraud patterns.

3.SYSTEM ANALYSIS

SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

- Operating System: Windows 7 Ultimate
- Programming Language: Python
- Front end: Html, Css, Javascript
- Framework: Django
- Data Base: MySQL (XAMPP Server).

3.2 HARDWARE REQUIREMENTS

- System Processor: core i3 or above
- Ram: 8 GB Minimum
- Hard disk: 512
- Key Board: Standard Windows Keyboard
- Mouse: Two or Three Button Mouse
- Monitor: SVGA

3.3 Functional Requirements

1. **Data Input:** The system should allow users to input historical e-commerce transaction data, including relevant parameters such as transaction amount, user behaviour patterns, timestamps, and payment methods .
2. **Model Training:** The system must be able to train using a **Support Vector Machine (SVM)** classifier and process mining techniques to enhance fraud detection .
3. **Accuracy Evaluation:** The system should provide a mechanism to evaluate the accuracy of fraud detection, using performance metrics such as **precision, recall, and F1-score** .
4. **Fraud Prediction:** Users should be able to request fraud detection analysis for new transactions based on the trained model .
5. **Model Tuning:** The system should allow users to fine-tune model parameters to improve fraud detection accuracy .

3.4 Non-Functional Requirements

1. **Performance:** The system should be capable of analysing transactions and detecting fraudulent activities efficiently, even for large datasets .
2. **Scalability:** The system should handle increasing volumes of transactional data without a significant drop in performance .
3. **Robustness:** The system should be able to manage missing or incomplete transaction data by using appropriate data imputation techniques or providing informative alerts .
4. **Maintainability:** The codebase should be well-documented, allowing developers to easily make updates and enhancements .
5. **Compatibility:** The system should support common data formats for e-commerce transactions and be exportable to standard reporting or visualisation tools .

3.5 SOFTWARE ENVIRONMENT

What is Python

Below are some facts about Python .

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms . Python programs generally are smaller than other programming languages like Java .
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time .
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc .

Advantages of Python

1. **Extensive Libraries:** Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more . So, we don't have to write the complete code for that manually .

2. **Extensible:** As we have seen earlier, Python can be extended to other languages . You can write some of your code in languages like C++ or C . This comes in handy, especially in projects .

3. **Improved Productivity:** The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do . Also, the fact that you need to write less and get more things done .

4. **Simple and Easy:** When working with Java, you may have to create a class to print 'Hello World' . But in Python, just a print statement will do . It is also quite easy to learn, understand, and code . This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java .

5. **Object-Oriented:** This language supports both the procedural and object-oriented programming paradigms . While functions help us with code reusability, classes and objects let us model the real world . A class allows the encapsulation of data and functions into one .

6. **Free and Open-Source:** Like we said earlier, Python is freely available . But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it . It downloads with an extensive collection of libraries to help you with your tasks .

7. **Portable:** When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform . But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere . This is called Write Once Run Anywhere (WORA) . However, you need to be careful enough not to include any system-dependent features .

8. Readable: Because it is not such a verbose language, reading Python is much like reading English . This is the reason why it is so easy to learn, understand, and code . It also does not need curly braces to define blocks, and indentation is mandatory **.** This further aids the readability of the code .

9. Interpreted: Lastly, we will say that it is an interpreted language . Since statements are executed one by one, debugging is easier than in compiled languages .

10. Embeddable: Complimentary to extensibility, Python is embeddable as well . You can put your Python code in your source code of a different language, like C++ . This lets us add scripting capabilities to our code in the other language .

ADVANTAGES OF PYTHON OVER OTHER LANGUAGES

1. Less Coding
2. Affordable and have built-in functions
3. Open source

What is Machine Learning

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't . Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush . The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data . Fundamentally, machine learning involves building mathematical models to help understand data . "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data . Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data . I'll leave to the reader the more philosophical 35 digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain . Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here .

Categories Of Machine Learning

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning . Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data . This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities . We will see examples of both types of supervised learning in the following section . Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself .

Need for Machine Learning

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems . We can call it data-driven decisions taken by machines, particularly to automate the process . These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently . The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale . That is why the need for machine learning arises .

Challenges in Machines Learning

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go . The reason behind is that ML has not been able to overcome number of challenges . The challenges that ML is facing currently are –

- Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges Use of low-quality data leads to the problems related to data preprocessing and feature extraction .
- Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.
- Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job .
- No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet .
- Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem .
- Curse of dimensionality − Another challenge ML model faces is too many features of data points . This can be a real hindrance .
- Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life .

Applications of Machine Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML . It is used to solve many real-world complex problems which cannot be solved with traditional approach . Following are some real-world applications of ML .

- Sentiment analysis
- Weather forecasting and prediction
- Speech recognition

- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping
- Customer segmentation
- Speech synthesis
- Stock market analysis and forecasting

Advantages of Machine learning

1. Easily identifies trends and patterns

2. No human intervention needed (automation)

3. Continuous Improvement

Terminologies of Machine Learning

1. **Model –** A model is a specific representation learned from data by applying some machine learning algorithm . A model is also called a hypothesis .
2. **Feature –** A feature is an individual measurable property of the data . A set of numeric features can be conveniently described by a feature vector . Feature vectors are fed as input to the model . For example, in order to predict a fruit, there may be features like color, smell, taste, etc .
3. **Target (Label) –** A target variable or label is the value to be predicted by our model . For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc .
4. **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on .
5. **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label) .

ADVANTAGES OF MACHINE LERNING

1. **Easily identifies trends and patterns:** Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans . For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them . It uses the results to reveal relevant advertisements to them .
2. **No human intervention needed (automation):** With ML, you don't need to babysit your project every step of the way . Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own . A common example of this is anti-virus software's they learn to filter new threats as they are recognized . ML is also good at recognizing spam .

3. **Continuous Improvement:** As ML algorithms gain experience, they keep improving in accuracy and efficiency . This lets them make better decisions . Say you need to make a weather forecast model . As the amount of data, you have keeps growing, your algorithms learn to make more accurate predictions faster .

4. **Handling multi-dimensional and multi-variety data:** Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments .

5. **Wide Applications:** You could be an e-tailer or a healthcare provider and make ML work for you . Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers .

DISADVANTAGES OF MACHINE LEARNING:

1. **Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality . There can also be times where they must wait for new data to be generated .

2. **Time and Resources**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy . It also needs massive resources to function . This can mean additional requirements of computer power for you .

3. **Interpretation of Results**

Another major challenge is the ability to accurately interpret results generated by the algorithms . You must also carefully choose the algorithms for your purpose .

4. **High error-susceptibility**

Machine Learning is autonomous but highly susceptible to errors . Suppose you train an algorithm with data sets small enough to not be inclusive . You end up with biased predictions coming from a biased training set . This leads to irrelevant advertisements being displayed to customers . In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time . And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it .

4.SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

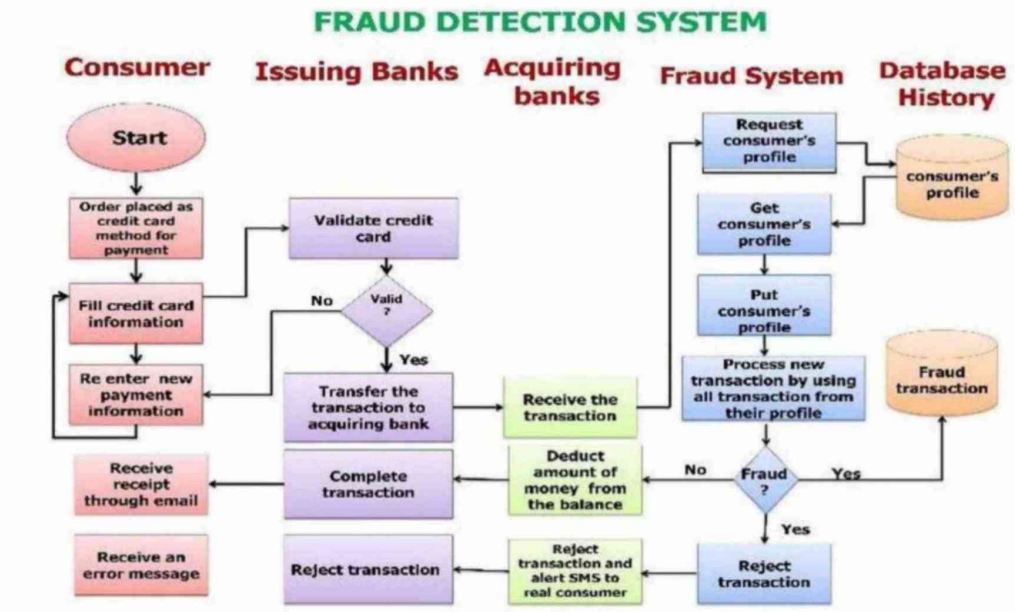


Figure: 4 .1 SYSTEM ARCHITECTURE

The **Fraud Detection System** for multi-participant e-commerce transactions is designed to identify and prevent fraudulent activities through an integrated approach involving machine learning and process mining techniques. The process begins when a consumer initiates a transaction by placing an order using their credit card, entering details such as the card number, CVV, expiration date, and billing address. This information is then validated by the issuing bank, which checks for the authenticity of the card, its expiration status, sufficient balance, and any prior fraudulent activity associated with it. If the validation fails, the consumer is prompted to enter alternative payment details. Upon successful verification, the transaction request moves to the acquiring bank, which processes the payment by deducting the amount from the consumer's balance and transferring it to the merchant's account. However, if the system detects any anomalies, the transaction is flagged for fraud analysis before being approved.

At the core of the fraud detection mechanism is a machine learning-based system that monitors transactions in real-time to identify suspicious activities . The system retrieves the consumer's profile from historical transaction records and analyses key attributes such as transaction amount, frequency, geographical location, and the type of merchant involved . Advanced feature engineering techniques extract meaningful data patterns, allowing the model to differentiate between normal and fraudulent transactions . Using classification algorithms like Support Vector Machine (SVM), the system evaluates each transaction, determining whether it aligns with past legitimate behaviour or exhibits fraudulent characteristics . If a transaction is flagged as fraudulent, an immediate alert is sent to the consumer via SMS or email for verification . In case of a confirmed fraud attempt, the transaction is blocked, preventing financial loss . On the other hand, if the transaction is classified as legitimate, it proceeds for final processing, ensuring a seamless experience for the consumer .

A crucial aspect of the system is its ability to continuously learn from new fraud patterns . The fraud detection database is regularly updated with newly identified fraudulent transactions, enhancing the system's accuracy in detecting emerging threats . This adaptive mechanism ensures that evolving fraud tactics are recognized, making the system robust against sophisticated cyber threats . Ultimately, by leveraging real-time monitoring, behavioural analysis, asnd predictive modelling, this fraud detection framework enhances the security of digital transactions, protecting both consumers and merchants from financial fraud .

4.2 UML Diagrams

UML, which stands for Unified Modeling Language, is a way to visually represent the architecture, design, and implementation of complex software systems. UML is a standardized modeling language that can be used across different programming languages and development processes, so the majority of software developers will understand it and be able to apply it to their work. When you're writing code, there are thousands of lines in an application, and it's difficult to keep track of the relationships and hierarchies within a software system. UML diagrams divide that software system into components and subcomponents.

4.2.1 Class Diagram: The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes *can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. Apart from this, each class may have certain "attributes" that uniquely identify the class.

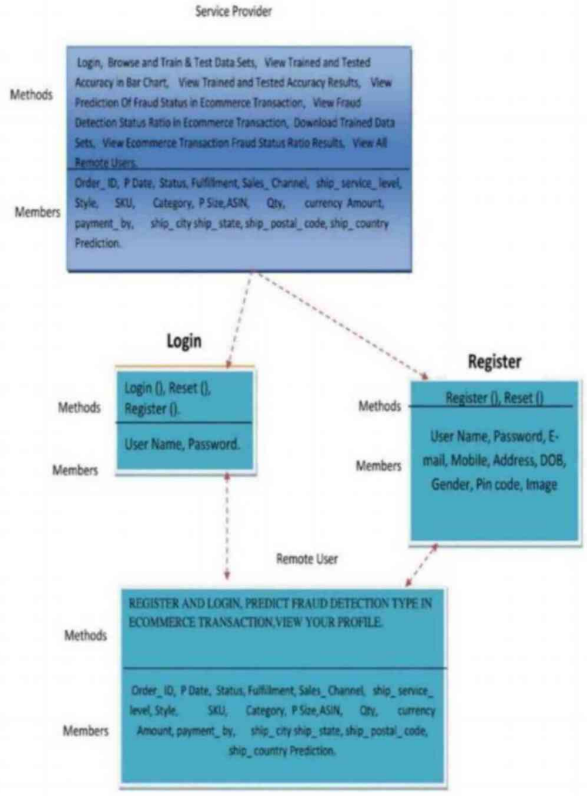


Figure: 4 .2 .1 CLASS DIAGRAM

4.2.2 Use Case Diagram: A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed. The actor can be a user, system etc.

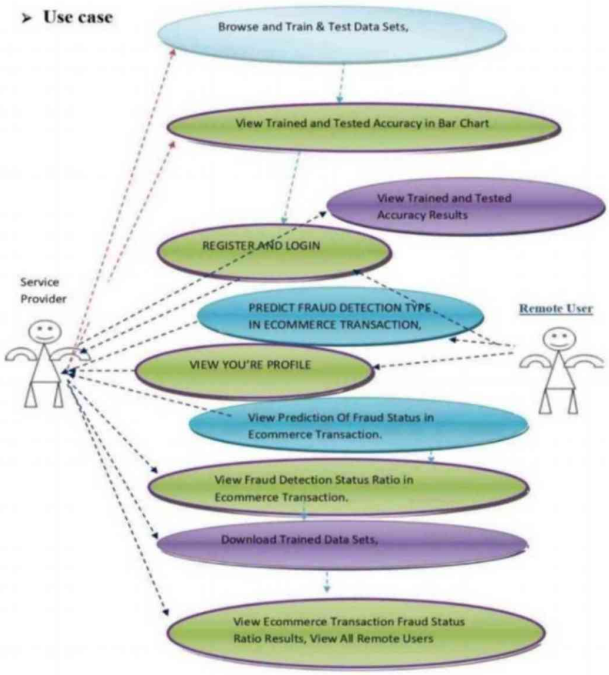


Figure: 4 .2 .2 USE CASE DIAGRAM

Actor: An actor is someone or something that: Interacts with or uses the system . Provides input to and receives information from the system . Is external to the system and has no control over the use cases . The actors are sometimes will be a "System" too . Actors are discovered by examining:

- Who directly uses the system?
- Who is responsible for maintaining the system?
- Who is using the system? Or, who is affected by the system?

Use case: A use case can be described as a specific way of using the system from a User's (actor's)Perspective . A more detailed description might characterize a use case as:

- Pattern of behavior the system exhibits
- A sequence of related transactions performed by an actor and the system
- Delivering something of value to the actor Use cases provide a means to:
- capture system requirements
- communicate with the end users and domain experts

  Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system .

  For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform . The use case should represent a course of events that leads to clear goal .

- Name the use cases .b
- Describe the use cases briefly by applying terms with which the user is familiar . This makes the description less ambiguous

Questions to identify use cases:

- What are the tasks of each actor?
- Will any actor create, store, change, remove or read information in the system? What use case will store, change, remove or read this information?
- Will any actor need to inform the system about sudden external changes? Does any actor need to inform about certain occurrences in the system? What use cases will support and maintains the system?

4.2.3 Activity diagram: In the figure graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency . In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system . An activity diagram shows the overall flow of control .

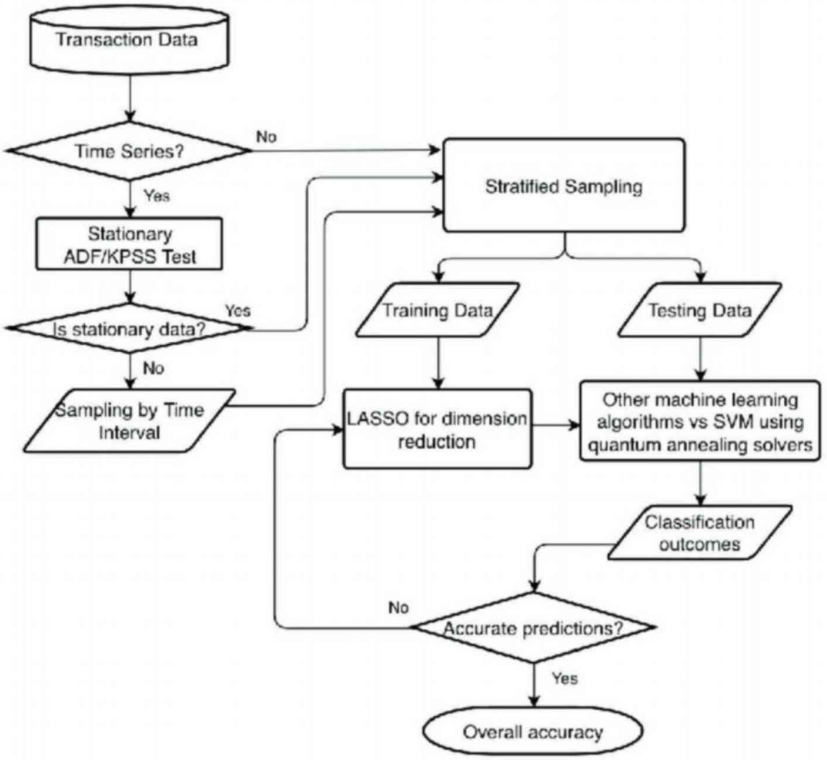


Figure: 4.2.3 ACTIVITY DIAGRAM

4.2.4 Sequence Diagram: The Sequence Diagram is a type of interaction diagram that illustrates how different processes operate and the order in which they interact . Sequence diagrams are also known as event diagrams, event scenarios, or timing diagrams. The sequence diagram visually represents the flow of messages between the objects or components involved in the fraud detection system. In the proposed system, the user interacts with the platform by initiating a transaction, which is then processed through various fraud detection mechanisms . The system collects transactional data, extracts relevant features using process mining techniques, and applies a Support Vector Machine (SVM)-based classification model to detect fraudulent behaviours . The sequence diagram illustrates the order in which these interactions take place, including data preprocessing, feature extraction, fraud analysis, and classification .

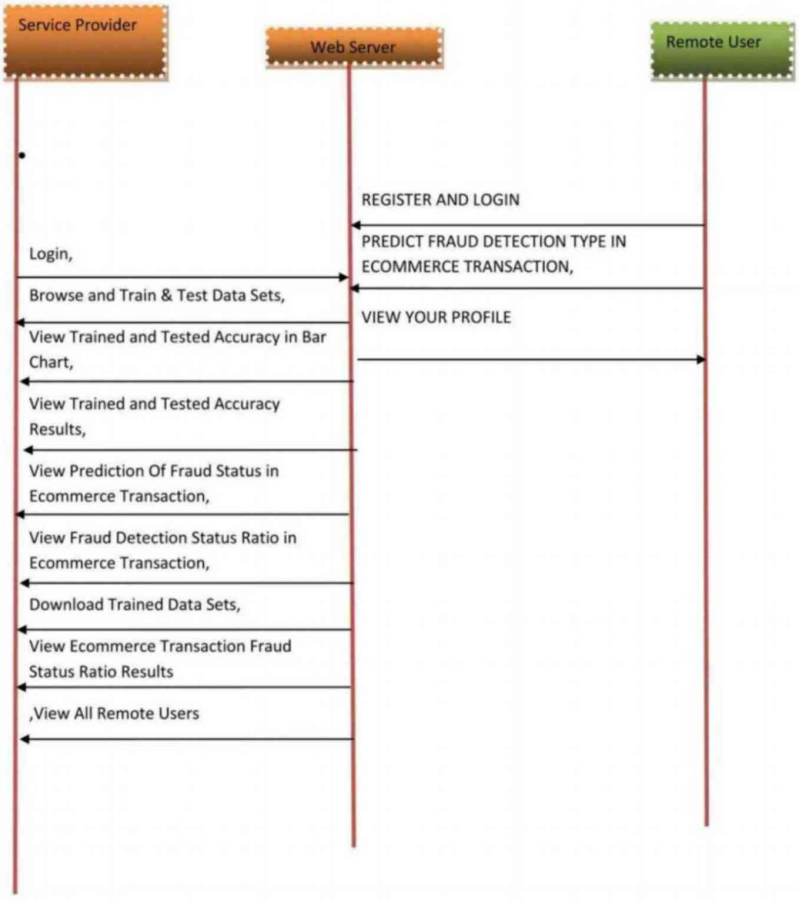


Figure: 4.2.4 SEQUENCE DIAGRAM

5.SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase, and a business proposal is put forth with a general plan for implementation, along with cost estimates . During the system analysis phase, the feasibility study of the proposed **Multi-Perspective Fraud Detection Method for Multi-Participant E-commerce Transactions** is conducted to ensure that the system is practical and beneficial . This analysis determines whether the proposed solution aligns with financial, technical, and social considerations, ensuring its successful deployment. Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

5.1 ECONOMICAL FEASIBILITY

This study evaluates the economic impact of implementing the fraud detection system . The financial resources available for the research, development, and deployment of the system are limited, making cost efficiency a crucial factor . The proposed system is economically feasible as it primarily relies on open-source machine learning libraries and process mining tools, reducing development costs . Only essential resources, such as cloud storage or additional computing power for training the fraud detection model, require investment . The system is designed to be cost-effective while ensuring high fraud detection accuracy, making it a viable solution for e-commerce platforms .

5.2 TECHNICAL FEASIBILITY

This study assesses whether the existing technical infrastructure can support the implementation of the fraud detection system . The proposed system is designed to work with standard e-commerce databases and platforms, ensuring minimal modifications are required for integration . The system utilises Python-based machine learning models and process mining techniques, which are compatible with most modern computing environments . Additionally, the computational requirements for fraud detection, such as training the Support Vector Machine (SVM) model, can be managed efficiently with cloud-based or on-premise resources . The system is technically feasible as it does not place excessive demands on the existing technological infrastructure .

5.3 SOCIAL FEASIBILITY

This aspect evaluates the acceptance of the fraud detection system by users, including e-commerce businesses and customers . The proposed system is designed to enhance transaction security without disrupting the user experience . Customers and merchants will benefit from a safer transaction environment, reducing financial losses due to fraud . Additionally, the system's transparency in fraud detection ensures that legitimate transactions are not unnecessarily blocked . Training and awareness programs can be conducted to familiarise stakeholders with the system's functionalities, ensuring smooth adoption . The system is socially feasible as it aims to build trust among users while maintaining a seamless e-commerce experience .

Effects of E-commerce Fraud

Introduction: E-commerce fraud affects businesses, consumers, and the overall digital economy . It leads to financial losses, damages trust in online transactions, and increases cybersecurity risks . Fraudulent activities in e-commerce include identity theft, payment fraud, account takeovers, and fake reviews . With the rapid growth of online shopping, fraudsters continuously adapt their tactics, making fraud detection and prevention crucial .

There are 2 two types of effects:

1. Short term effects
2. Long term effects

1. Short Term Effects: Short-term effects are immediate consequences that impact businesses and customers directly . These include financial losses, chargebacks, and temporary account suspensions . Fraud can also result in customer frustration and decreased confidence in online transactions . Businesses may experience increased operational costs to investigate fraudulent activities and handle customer disputes .

Some of the Effects given below:

- Unauthorized transactions
- Chargebacks and refund requests
- Customer dissatisfaction
- Temporary account bans or suspensions
- Increased operational costs
- Loss of sensitive data

2. Long Term Effects: Long-term effects of e-commerce fraud can be more severe, impacting the reputation and sustainability of businesses . Fraud-related losses can lead to financial instability, legal actions, and regulatory penalties . Customers may lose trust in online platforms, leading to reduced sales and lower customer retention rates . Businesses may also face increased security and compliance costs to prevent future fraud .

Some long-term effects include:

- Reputation damage
- Loss of customer trust
- Financial instability
- Legal consequences and regulatory fines
- Increased cybersecurity and compliance costs
- Decrease in overall e-commerce growth

Both short-term and long-term exposure to e-commerce fraud can have serious consequences . It is essential for businesses and consumers to adopt preventive measures, such as multi-factor authentication, fraud detection systems, and secure payment gateways . Raising awareness and implementing robust security practices can help mitigate the risks associated with e-commerce fraud and ensure a safer digital shopping experience .

6.IMPLEMENTATION

6.1 MODULES USED IN PROJECT

Loading Dataset and Data Preprocessing

1. pandas: Pandas is a popular open-source Python library used for data manipulation and analysis . It provides easy-to-use data structures and functions for efficiently working with structured data, such as tabular data and more . Use pandas to read the dataset containing features related to air quality parameters like pollutant concentrations, meteorological conditions, and geographical factors . Pandas can help in handling missing values, outliers, and inconsistencies in the dataset, ensuring data quality before training the machine learning model . Use pandas in conjunction with visualization libraries like Matplotlib or Seaborn to create visualizations such as bar graphs to explore the relationships between features and the target variable . The import is used to import the pandas:

```
import pandas as pd
```

Automating Calculations: (Using built-in functions)

2. numpy: Numpy is a fundamental Python library for numerical computing, providing support for multi-dimensional arrays and matrices . It offers a wide range of mathematical functions for operations like linear algebra etc . After model training, numpy arrays are used to represent test data for making predictions with trained models, enabling efficient computation of predicted AQI values or class labels . Numpy aids in calculating performance metrics such as Root Mean Squared Error (RMSE) to evaluate the performance of the trained models .

```
import numpy as np
```

Data Visualization:

3. seaborn: Seaborn is particularly useful for visualizing statistical relationships in datasets, as it offers specialized functions for exploring patterns, trends, and correlations between variables . It also provides support for categorical data visualization and supports the use of color palettes to enhance the aesthetics of plots . Seaborn's pairplot function allows for the creation of pairwise scatter plots to explore relationships between features and the target variable, aiding in feature selection and understanding feature importance . Seaborn's barplot function can be used to visualize cross-validation scores across different folds or parameters.

Seaborn facilitates the comparison of multiple models by creating side-by-side box plots or violin plots.

```
import seaborn as sns
```

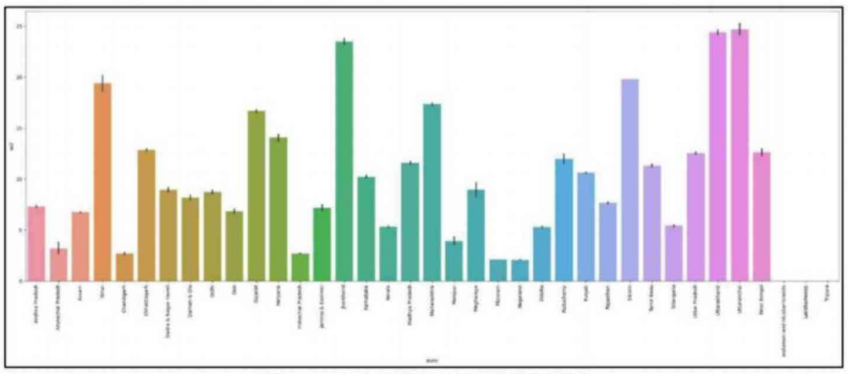


Figure: 6 .1 .1 SEABORN GRAPH

4. matplotlib: Matplotlib is a versatile Python library for creating static, interactive, and high-quality visualizations .matplotlib .pyplot is a module to plot the graphs . It offers extensive customization options for various plot types, supports multiple backend rendering engines, and can be used standalone or with other data Matplotlib in an Air Quality Index (AQI) prediction project with Decision Tree models aids in visualizing data distributions . Cross-validation scores, error analysis, and model comparison are showcased using Matplotlib's diverse plotting capabilities .

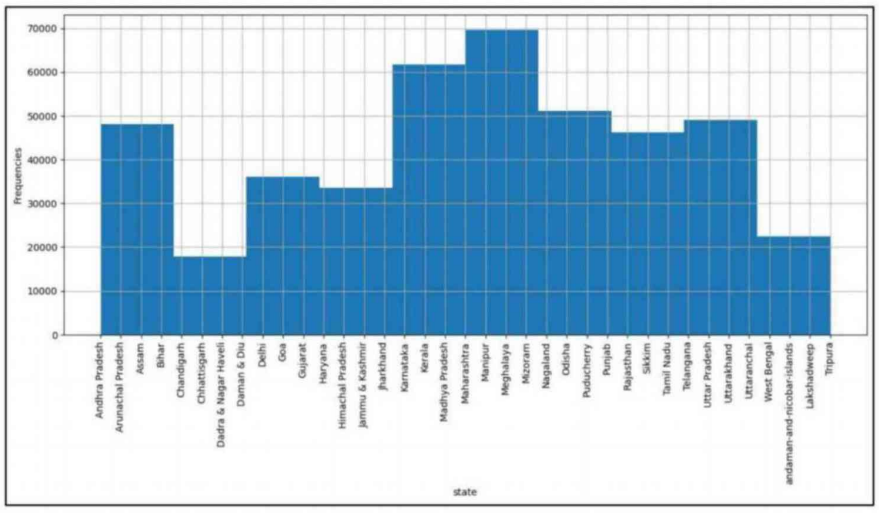


Figure: 6 .1 .2 MATPLOTLIB GRAPH

5. Scikitlearn: Scikit-learn is a prominent Python library for machine learning tasks, offering a wide array of algorithms for classification, regression, clustering, and more . Its intuitive interface facilitates the implementation of machine learning models, data preprocessing, and model evaluation with ease . Compatible with other Python libraries like NumPy and Pandas . The scikit learn is used to split dataset into training and testing data . By importing train_test_split module we can split the dataset .

```
import scikit-learn as sklearn
```

from sklearn .model_selection import train_test_split

Importing Algorithms from scikit learn

The Decision Tree Regressor and Decision Tree Classifier algorithms, imported from scikit-learn, are powerful tools for predictive modeling . Decision Tree Regressor builds a tree- like structure to predict target variables, while Decision Tree Classifier predicts categorical target variables by partitioning the feature space into regions . These algorithms are non- parametric and handle non-linear relationships well . They are interpretable, allowing users to understand the decision-making process behind predictions .

from sklearn.tree import DecisionTreeRegressor

The Decision Trees are used to build the tree like structure and also used to handle the complex data . The above import statement is used to import the DecisionTreeRegressor .

Importing Evaluation metrics from scikit learn

Scikit-learn provides a comprehensive suite of evaluation metrics for assessing the performance of machine learning models . Common regression metrics include Root Mean Squared Error (RMSE), and R-squared score, while classification metrics encompass accuracy, precision etc . Users can easily compare models and optimize performance using these evaluation metrics, ensuring robustness and reliability in machine learning applications . Utilize metrics like accuracy, precision to evaluate the performance of Decision Tree Classifier models in predicting AQI categories (e .g ., good, moderate, unhealthy) .

from sklearn import metrics
from sklearn .metrics import mean_absolute_error,mean_squared_error,r2_score
from sklearn .metrics import accuracy_score

The sklearn .metrics module is used to check the model's accuracy . Model's accuracy is very important to predict the correct result for the given data the user should receive the accurate data or output for the given input.

6.2 ALGORITHMS

1. **SUPPORT VECTOR MACHINE:**

   Support Vector Machine (SVM) algorithm is a type of supervised machine learning model that aims to find the best hyperplane to separate classes in a dataset. SVM works by mapping the input data into a higher-dimensional space, where it becomes easier to find a linear separation between classes. The algorithm then identifies the support vectors, which are the data points that lie closest to the hyperplane and have the greatest influence on its position.

   How We Used SVM in Our Project?

   In our project, SVM plays a crucial role in the final classification stage. Here's how we incorporated it:

1. **Feature Extraction from Event Logs:**

   We collected transaction event logs from an e-commerce platform.

   Key behavioral features such as **transaction frequency, device used, order value, IP address history, and past fraud reports** were extracted.

2. **Data Preprocessing:**

   Missing values were handled, and categorical data (e.g., payment mode, device type) was converted into numerical values using **one-hot encoding**.

   Data was **normalised** to ensure that all features contribute equally to the classification.

3. **Training the SVM Model:**

   The extracted features were fed into an **SVM classifier** to learn the patterns of fraud.

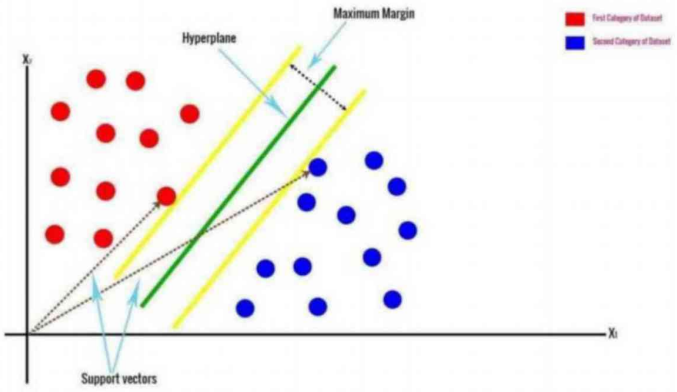
   A **Radial Basis Function (RBF) kernel** was used to capture complex relationships between fraudulent and non-fraudulent transactions.

   We used **grid search** to optimise hyperparameters like C (regularisation parameter) and gamma (kernel coefficient).

4. **Fraud Detection in Real-Time:**

   Once trained, the SVM model was deployed to **analyse transactions dynamically**.

   If a transaction was classified as **fraudulent**, it was flagged for further review or blocked automatically.



2. **LOGISTIC REGRESSION:** Logistic Regression is a type of supervised machine learning algorithm used for binary classification problems, where the goal is to predict the probability of an event occurring or not occurring. The algorithm works by learning a logistic function that maps the input features to a probability between 0 and 1, indicating the likelihood of the positive class. Logistic Regression uses a sigmoid function to model the probability, and the output is typically interpreted as a probability or a binary classification (0 or 1).

   How We Used Logistic Regression in Our Project?

   Logistic Regression was incorporated into our fraud detection model as a baseline classifier for **early-stage fraud analysis** before integrating advanced techniques like SVM.

   Step 1: Feature Engineering

   We extracted essential features from e-commerce **transaction event logs**, such as:

- **Transaction Amount** – Unusually high or low amounts.
- **Transaction Frequency** – Repeated purchases within a short time frame.
- **IP Address and Location Mismatch** – A mismatch between registered and transaction locations.
- **Device Used** – Whether the user frequently changes devices.
- **Payment Method** – Use of high-risk payment methods like cryptocurrency.

   These features were **numerically encoded** and **normalised** to fit the model.

   Step 2: Training the Logistic Regression Model

1. We split the dataset into **training and testing sets** (e.g., 80% training, 20% testing).
2. We applied **Logistic Regression** to classify transactions as fraud or non-fraud.
3. **Regularisation (L1 or L2)** was used to prevent overfitting and improve model.

4. The model was trained using **Gradient Descent** to minimise the **Log Loss function**.

Step 3: Fraud Probability Scoring

- Logistic Regression provided a **fraud probability score** (0 to 1) for each transaction.
- Transactions exceeding a defined **threshold (e.g., 0.7)** were flagged for further review

3. DECISION TREE CLASSIFIER:

A Decision Tree Classifier is a type of supervised machine learning algorithm that uses a tree like model to classify data into different classes. The algorithm works by recursively partitioning the data into smaller subsets based on the values of the input features. Each internal node in the tree represents a feature or attribute, and the branches represent the possible values of that feature. The leaf nodes represent the predicted class labels. The decision tree classifier learns to split the data at each node by selecting the feature that best separates the classes, and the process is repeated until a stopping criterion is reached. The resulting tree can be used to classify new, unseen data by traversing the tree from the root node to a leaf node, following the decisions made at each internal node.

How We Used Decision Tree in Our Project?

The **Decision Tree Classifier** was used to detect fraudulent transactions based on multiple behavioral and transactional features.

Step 1: Feature Selection & Engineering

We extracted important fraud indicators such as:

- **Transaction Amount** – Unusually high or low amounts.
- **Transaction Frequency** – Number of transactions within a short period.
- **IP Address & Location Mismatch** – Whether the user's location changes frequently.
- **Device Used** – Checking for multiple device changes in a short span.
- **Payment Method** – Risky payment options like cryptocurrency.

These features were used as **decision nodes** in the tree, helping to classify whether a transaction is fraudulent or legitimate.

Step 2: Building the Decision Tree Model

1. The dataset was **split into training and testing sets** (e.g., 80% training, 20% testing).
2. A **Gini Impurity or Entropy criterion** was used to decide the best splits.

3. The model was trained using a **top-down recursive approach**, where the dataset was split at each node based on the most important feature.
4. **Pruning techniques** (e.g., **max depth, min samples split**) were applied to prevent overfitting.

Step 3: Fraud Classification & Real-Time Detection

- The Decision Tree classified transactions based on predefined **rules** (e.g., "If transaction amount > ₹50,000 & IP mismatch → Fraud").

5. K-Nearest Neighbors (KNN) :

K-Nearest Neighbors (**KNN**) is a **non-parametric, instance-based** learning algorithm that classifies a transaction based on the similarity to its nearest neighbors. It is useful for fraud detection because:

- It **does not assume a specific data distribution**, making it flexible.
- It can **detect anomalies** by comparing transactions with past fraudulent and non-fraudulent cases.
- It is effective in **high-dimensional spaces**, where multiple transaction features are analyzed.

How We Used KNN in Our Project?

KNN was used as a classification model to detect fraudulent transactions by **comparing them with historical transaction patterns**.

Step 1: Feature Extraction & Preprocessing

We extracted key transaction features such as:

- **Transaction Amount** – Identifies outliers (unusually high or low amounts).
- **Time Interval Between Transactions** – Shorter time gaps may indicate fraud.
- **User Behavior Similarity** – Checks whether the transaction behavior deviates from past activities.
- **IP & Device Consistency** – Compares with previous transactions for anomalies.

To ensure accurate distance measurement, **feature scaling** (e.g., Min-Max Scaling or Standardization) was applied.

Step 2: Training the KNN Model

1. **Choosing the Optimal K-value:**
   - The **elbow method** was used to find the best K value, balancing bias and variance.
   - Typically, a smaller K detects anomalies better, while a larger K smooths classification.
2. **Distance Metric Selection:**
   - We used **Euclidean distance** to measure transaction similarity.
   - In cases where features had different units, **Manhattan distance** was tested.
3. **Model Training & Testing:**
   - The dataset was split into **training and testing sets** (e.g., 80% training, 20% testing).
   - Each new transaction was classified based on the **majority label** of its nearest K transactions.

Step 3: Real-Time Fraud Detection

- If a transaction had **more fraudulent neighbors than legitimate ones**, it was flagged as **fraudulent**.
- The fraud probability was based on the ratio of fraudulent transactions in the nearest K neighbors.

5. RANDOM FOREST :

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

How We Used Random Forest in Our Project?

Random Forest was used to classify transactions as **fraudulent or legitimate** based on user behaviour and transaction patterns.

Step 1: Feature Selection & Engineering

We extracted key fraud detection indicators such as:

- **Transaction Amount** – Higher-than-usual amounts flagged as suspicious.
- **Transaction Frequency** – Rapid multiple transactions may indicate fraud.
- **Device & IP Address Consistency** – Checking if the user frequently switches devices or locations.
- **Historical Fraud Labels** – Patterns from past fraudulent transactions.

Random Forest also includes **feature importance ranking**, helping us identify the most significant fraud detection factors.

Step 2: Model Training & Testing

1. **Dataset Splitting:**
   - The data was split into **training (80%) and testing (20%) sets**.
2. **Building the Random Forest Model:**
   - We set the **number of trees (n_estimators)** to 100-200 for optimal results.
   - Used **Gini Impurity or Entropy** as the splitting criterion.
   - Applied **Bootstrapping** to create diverse subsets for training individual trees.
3. **Ensemble Voting Mechanism:**
   - Each decision tree in the forest **votes** on whether a transaction is fraudulent.
   - The final classification is determined by **majority voting** across all trees.

Step 3: Fraud Detection in Real-Time Transactions

- Each incoming transaction was processed through multiple trees.

7.SOURCE CODE

#Importing necessary libraries

```
from django .db .models import  Count, Avg
from django .shortcuts import render, redirect
from django .db .models import Count
from django .db .models import Q
import datetime
import xlwt
from django .http import HttpResponse

import pandas as pd
from sklearn .feature_extraction .text import CountVectorizer
from sklearn .metrics import accuracy_score, confusion_matrix, classification_report
from sklearn .metrics import accuracy_score
from sklearn .tree import DecisionTreeClassifier

# Create your views here .
from Remote_User .models import
ClientRegister_Model,fraud_detection,detection_ratio,detection_accuracy

def serviceproviderlogin(request):
    if request .method  == "POST":
        admin = request .POST .get('username')
        password = request .POST .get('password')
        if admin == "Admin" and password =="Admin":
            detection_accuracy .objects .all() .delete()
            return redirect('View_Remote_Users')

    return render(request,'SProvider/serviceproviderlogin .html')

def View_Fraud_Detection_Ratio(request):
    detection_ratio .objects .all() .delete()
    ratio = ""
    kword = 'Fraud Found in ECommerce Transaction'
    print(kword)
    obj = fraud_detection .objects .all() .filter(Q(Prediction=kword))
    obj1 = fraud_detection .objects .all()
    count = obj .count();
    count1 = obj1 .count();
    ratio = (count / count1) * 100
    if ratio != 0:
```

```
        detection_ratio .objects .create(names=kword, ratio=ratio)

    ratio12 = ""
    kword12 = 'No Fraud Found in ECommerce Transaction'
    print(kword12)
    obj12 = fraud_detection .objects .all() .filter(Q(Prediction=kword12))
    obj112 = fraud_detection .objects .all()
    count12 = obj12 .count();
    count112 = obj112 .count();
    ratio12 = (count12 / count112) * 100
    if ratio12 != 0:
        detection_ratio .objects .create(names=kword12, ratio=ratio12)


    obj = detection_ratio .objects .all()
    return render(request, 'SProvider/View_Fraud_Detection_Ratio .html', {'objs': obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model .objects .all()
    return render(request,'SProvider/View_Remote_Users .html',{'objects':obj})

def charts(request,chart_type):
    chart1 = detection_ratio .objects .values('names') .annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts .html", {'form':chart1, 'chart_type':chart_type})

def charts1(request,chart_type):
    chart1 = detection_accuracy .objects .values('names') .annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts1 .html", {'form':chart1, 'chart_type':chart_type})

def View_Fraud_Detection_Status(request):
    obj =fraud_detection .objects .all()
    return render(request, 'SProvider/View_Fraud_Detection_Status .html', {'list_objects': obj})

def likeschart(request,like_chart):
    charts =detection_accuracy .objects .values('names') .annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/likeschart .html", {'form':charts, 'like_chart':like_chart})

def Download_Trained_DataSets(request):

    response = HttpResponse(content_type='application/ms-excel')
    # decide file name
    response['Content-Disposition'] = 'attachment; filename="Predicted_Datasets .xls"'
    # creating workbook
    wb = xlwt .Workbook(encoding='utf-8')
```

```
    # adding sheet
    ws = wb .add_sheet("sheet1")
    # Sheet header, first row
    row_num = 0
    font_style = xlwt .XFStyle()
    # headers are bold
    font_style .font .bold = True
    # writer = csv .writer(response)
    obj = fraud_detection .objects .all()
    data = obj  # dummy method to fetch data .
    for my_row in data:
        row_num = row_num + 1

        ws .write(row_num, 0, my_row .Order_ID, font_style)
        ws .write(row_num, 1, my_row .PDate, font_style)
        ws .write(row_num, 2, my_row .Status, font_style)
        ws .write(row_num, 3, my_row .Fulfilment, font_style)
        ws .write(row_num, 4, my_row .Sales_Channel, font_style)
        ws .write(row_num, 5, my_row .ship_service_level, font_style)
        ws .write(row_num, 6, my_row .Style, font_style)
        ws .write(row_num, 7, my_row .SKU, font_style)
        ws .write(row_num, 8, my_row .Category, font_style)
        ws .write(row_num, 9, my_row .PSize, font_style)
        ws .write(row_num, 10, my_row .ASIN, font_style)
        ws .write(row_num, 11, my_row .Qty, font_style)
        ws .write(row_num, 12, my_row .currency, font_style)
        ws .write(row_num, 13, my_row .Amount, font_style)
        ws .write(row_num, 14, my_row .payment_by, font_style)
        ws .write(row_num, 15, my_row .ship_city, font_style)
        ws .write(row_num, 16, my_row .ship_state, font_style)
        ws .write(row_num, 17, my_row .ship_postal_code, font_style)
        ws .write(row_num, 18, my_row .ship_country, font_style)
        ws .write(row_num, 19, my_row .Prediction, font_style)

wb .save(response)
return response

def train_model(request):
detection_accuracy .objects .all() .delete()

df = pd .read_csv('Datasets .csv')

def apply_response(Label):
if (Label == 0):
```

```
return 0  # No Fraud Found
elif (Label == 1):
return 1  #  Fraud Found

df['Label'] = df['Label'] .apply(apply_response)

cv = CountVectorizer()
X = df['Order_ID']
y = df['Label']

print("Order_ID")
print(X)
print("Label")
print(y)

cv = CountVectorizer()
X = cv .fit_transform(X)

   models = []
   from sklearn .model_selection import train_test_split
   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0 .20)
   X_train .shape, X_test .shape, y_train .shape

   print(X_test)

   print("Naive Bayes")

   from sklearn .naive_bayes import MultinomialNB
   NB = MultinomialNB()
   NB .fit(X_train, y_train)
   predict_nb = NB .predict(X_test)
   naivebayes = accuracy_score(y_test, predict_nb) * 100
   print(naivebayes)
   print(confusion_matrix(y_test, predict_nb))
   print(classification_report(y_test, predict_nb))
   models .append(('naive_bayes', NB))
   detection_accuracy .objects .create(names="Naive Bayes", ratio=naivebayes)

   # SVM Model
   print("SVM")
   from sklearn import svm
   lin_clf = svm .LinearSVC()
   lin_clf .fit(X_train, y_train)
   predict_svm = lin_clf .predict(X_test)
```

```
    svm_acc = accuracy_score(y_test, predict_svm) * 100
    print(svm_acc)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, predict_svm))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, predict_svm))
    models .append(('svm', lin_clf))
    detection_accuracy .objects .create(names="SVM", ratio=svm_acc)

    print("Logistic Regression")

    from sklearn .linear_model import LogisticRegression
    reg = LogisticRegression(random_state=0, solver='lbfgs') .fit(X_train, y_train)
    y_pred = reg .predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, y_pred) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, y_pred))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, y_pred))
    models .append(('logistic', reg))
detection_accuracy .objects .create(names="Logistic Regression",
ratio=accuracy_score(y_test,     y_pred) * 100)

    print("Decision Tree Classifier")
    dtc = DecisionTreeClassifier()
    dtc .fit(X_train, y_train)
    dtcpredict = dtc .predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, dtcpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, dtcpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, dtcpredict))
    models .append(('DecisionTreeClassifier', dtc))
detection_accuracy .objects .create(names="Decision Tree Classifier",
ratio=accuracy_score(y_test, dtcpredict) * 100)

    print("Extra Tree Classifier")
    from sklearn .tree import ExtraTreeClassifier
    etc_clf = ExtraTreeClassifier()
    etc_clf .fit(X_train, y_train)
    etcpredict = etc_clf .predict(X_test)
    print("ACCURACY")
```

```
    print(accuracy_score(y_test, etcpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, etcpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, etcpredict))
    models .append(('RandomForestClassifier', etc_clf))
  detection_accuracy .objects .create(names="Extra Tree Classifier",
  ratio=accuracy_score(y_test, etcpredict) * 100)

    csv_format = 'Results .csv'
    df .to_csv(csv_format, index=False)
    df .to_markdown

    obj = detection_accuracy .objects .all()
    return render(request,'SProvider/train_model .html', {'objs': obj})

from django.db import models

# Create your models here.

from django.db.models import CASCADE

class ClientRegister_Model(models.Model):

   username = models.CharField(max_length=30)
   email = models.EmailField(max_length=30)
   password = models.CharField(max_length=10)
   phoneno = models.CharField(max_length=10)
   country = models.CharField(max_length=30)
   state = models.CharField(max_length=30)
   city = models.CharField(max_length=30)
   gender= models.CharField(max_length=30)
   address= models.CharField(max_length=30)

class fraud_detection(models.Model):

   Order_ID= models.CharField(max_length=300)
   PDate= models.CharField(max_length=300)
   Status= models.CharField(max_length=300)
   Fulfilment= models.CharField(max_length=300)
   Sales_Channel= models.CharField(max_length=300)
   ship_service_level= models.CharField(max_length=300)
   Style= models.CharField(max_length=300)
   SKU= models.CharField(max_length=300)
```

```
    Category= models.CharField(max_length=300)
    PSize= models.CharField(max_length=300)
    ASIN= models.CharField(max_length=300)
    Qty= models.CharField(max_length=300)
    currency= models.CharField(max_length=300)
    Amount= models.CharField(max_length=300)
    payment_by= models.CharField(max_length=300)
    ship_city= models.CharField(max_length=300)
    ship_state= models.CharField(max_length=300)
    ship_postal_code= models.CharField(max_length=300)
    ship_country= models.CharField(max_length=300)
    Prediction= models.CharField(max_length=300)

class detection_accuracy(models.Model):

    names = models.CharField(max_length=300)
    ratio = models.CharField(max_length=300)

class detection_ratio(models.Model):

    names = models.CharField(max_length=300)
    ratio = models.CharField(max_length=300)
```

7.1 Dataset used in the project: A snapshot of dataset is shown below . The dataset contains Order_ID, PDate, Status, Fulfilment, Sales_Channel, ship_service_level, Style, SKU, Category, PSize, ASIN, QTY, currency, Amount, paymrnt_by, ship_city, ship_state, ship_postal_code, ship_country, Label .

Order_ID	PDate	Status	Fulfilment	Sales_Chann	ship_service_	Style	SKU	Category	PSize	ASIN	Qty	currency	Amount	payment_by	ship_city	ship_state	ship_postal-c	ship_country	Label
182.22.31.2	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-N	Set	S	B09KXVBD7	0	INR	647.62	Credit Card	MUMBAI	MAHARASHT	400081	IN	1
172.217.3.1	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-	kurta	3XL	B09K3WF53	1	INR	406	Credit Card	BENGALURU	KARNATAKA	560085	IN	1
10.42.0.211	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-	kurta	XL	B07WV4JV4I	1	INR	329	Credit Card	NAVI MUMBI	MAHARASHT	410210	IN	0
172.217.12.	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dre	L	B099NRCT7E	0	INR	753.33	Credit Card	PUDUCHERR	PUDUCHERR	605008	IN	1
140.205.32.	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-	Top	3XL	B098714BZP	1	INR	574	Credit Card	CHENNAI	TAMIL NADU	600073	IN	0
203.205.147	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET264	SET264-KR-N	Set	XL	B08YN7XDSC	1	INR	824	Debit card	GHAZIABAD	UTTAR PRADI	201102	IN	0
10.42.0.151	04-30-22	Shipped	Amazon	Amazon.in	Expedited	J0095	J0095-SET-L	Set	L	B08CMHNW	1	INR	653	Credit Card	CHANDIGARI	CHANDIGARI	160036	IN	1
10.42.0.211	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-	kurta	S	B081WX4G4	1	INR	399	Credit Card	HYDERABAD	TELANGANA	500032	IN	1
172.217.12.	04-30-22	Cancelled	Amazon	Amazon.in	Expedited	SET200	SET200-KR-N	Set	3XL	B08L91ZZXN	0			Credit Card	HYDERABAD	TELANGANA	500008	IN	0
10.42.0.211	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3461	JNE3461-KR-	kurta	XXL	B08B3XF5MI	1	INR	363	Credit Card	Chennai	TAMIL NADU	600041	IN	1
172.217.10.	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3160	JNE3160-KR-	kurta	S	B07K3YQLF1	1	INR	685	Credit Card	CHENNAI	TAMIL NADU	600073	IN	0
10.42.0.1-1(	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3500	JNE3500-KR-	kurta	XS	B098117DJ3	1	INR	364	Credit Card	NOIDA	UTTAR PRADI	201303	IN	1
10.42.0.211	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-	kurta	XS	B081XCMYX	1	INR	399	Credit Card	Amravati.	MAHARASHT	444606	IN	0
10.42.0.42	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET182	SET182-KR-D	Set	XS	B085HS9477	1	INR	657	Credit Card	MUMBAI	MAHARASHT	400053	IN	1
10.42.0.211	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	J0351	J0351-SET-L	Set	L	B09CSSQN4F	1	INR	771	Credit Card	MUMBAI	MAHARASHT	400053	IN	0
10.42.0.211	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	PJNE3368	PJNE3368-KI	kurta	6XL	B09PY995VJ	1	INR	544	Debit card	GUNTAKAL	ANDHRA PRA	515801	IN	0
172.217.11.	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3721	JNE3721-KR-	kurta	XXL	B099FCT65D	1	INR	329	Debit card	JAIPUR	RAJASTHAN	302020	IN	1
203.205.158	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3405	JNE3405-KR-	kurta	XL	B081WT6GG	1	INR	399	Debit card	NEW DELHI	DELHI	110074	IN	0
172.217.6.2	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	JNE3697	JNE3697-KR-	kurta	XXL	B098133PV5	1	INR	458	Debit card	Gurgaon	HARYANA	122004	IN	1
10.42.0.211	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET254	SET254-KR-N	Set	XS	B0983DDPL(	1	INR	886	Credit Card	BENGALURU	KARNATAKA	560017	IN	1
192.229.163	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3795	JNE3795-KR-	kurta	3XL	B09HMXJVF5	1	INR	517	Credit Card	TIRUCHIRAPI	TAMIL NADU	620018	IN	1
10.42.0.211	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET345	SET345-KR-N	Set	M	B09KXV4BN(	1	INR	666	Credit Card	BENGALURU	KARNATAKA	560040	IN	0
209.10.120.	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3373	JNE3373-KR-	kurta	L	B082W7GVF	1	INR	376	Credit Card	HYDERABAD	TELANGANA	500072	IN	0
172.217.11.	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET291	SET291-KR-P	Set	M	B099NK55YC	0	INR	570.48	Credit Card	pune	MAHARASHT	411044	IN	0
172.217.3.1	04-30-22	Shipped	Amazon	Amazon.in	Expedited	MEN5002	MEN5002-KF	kurta	L	B08YYYDN9R	1	INR	499	Credit Card	TEZPUR	ASSAM	784001	IN	1
172.217.10.	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	NW030	NW030-TP-P	Set	XS	B09G2RQSRZ	1	INR	582	Credit Card	RANCHI	JHARKHAND	834002	IN	1
172.217.12.	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	JNE3415	JNE3415-KR-	kurta	3XL	B082W8JXJ5	1	INR	299	Credit Card	BILASPUR	CHHATTISGA	495001	IN	1
172.217.12.	04-30-22	Shipped	Amazon	Amazon.in	Expedited	PJNE2199	PJNE2199-KI	kurta	4XL	B09LD2W9X	1	INR	459	Credit Card	PUNE	MAHARASHT	411052	IN	1
222.73.28.9	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3567	JNE3567-KR-	kurta	M	B08KRXV1QI	1	INR	399	Credit Card	BENGALURU	KARNATAKA	560037	IN	1
10.42.0.151	04-30-22	Cancelled	Merchant	Amazon.in	Standard	JNE2132	JNE2132-KR-	kurta	3XL	B07JG3CND(	0			Credit Card	GUWAHATI	ASSAM	781003	IN	0
10.42.0.151	04-30-22	Shipped	Amazon	Amazon.in	Expedited	J0341	J0341-DR-S	Western Dre	S	B099NR761	1	INR	791	Credit Card	THIRUVARUR	TAMIL NADU	613704	IN	1
10.42.0.211	04-30-22	Shipped	Amazon	Amazon.in	Expedited	MEN5009	MEN5009-KF	kurta	XL	B08YYTCPYX	1	INR	499	Debit card	LUCKNOW	UTTAR PRADI	226010	IN	0
10.42.0.151	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	J0011	J0011-LCD-N	Set	M	B08B3YNJG5	1	INR	1233	Debit card	VISAKHAPATI	ANDHRA PRA	530016	IN	1
172.217.10.	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	JNE3766	JNE3766-KR-	kurta	M	B09K3XR43\	1	INR	517	Debit card	JEYPUR	ODISHA	764001	IN	0
182.254.5.2	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3373	JNE3373-KR-	kurta	XL	B082W8BXV	1	INR	376	Debit card	HYDERABAD	TELANGANA	500028	IN	1
10.42.0.151	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	MEN5001	MEN5001-KF	kurta	XL	B08YYRH2QE	1	INR	499	Debit card	LUCKNOW	UTTAR PRADI	226016	IN	0
192.229.17!	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET345	SET345-KR-N	Set	L	B09KXT4VG7	1	INR	666	Credit Card	CHENNAI	TAMIL NADU	600033	IN	1
10.42.0.151	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3787	JNE3787-KR-	kurta	S	B09RKBXM5	1	INR	487	Credit Card	NEW DELHI	DELHI	110092	IN	0
10.42.0.211	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3543	JNE3543-KR-	kurta	S	B08HHJP41L	1	INR	368	Credit Card	NEW DELHI	DELHI	110092	IN	1
217.69.136.	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3405	JNE3405-KR-	kurta	L	B081WSCKP	1	INR	399	Debit card	THIRUVANAN	KERALA	695011	IN	0
10.42.0.211	04-30-22	Shipped	Amazon	Amazon.in	Expedited	J0211	J0211-DR-XX	Ethnic Dress	XXL	B09831VWD	1	INR	699	Debit card	GREATER NOI	UTTAR PRADI	201306	IN	1
180.76.152.	04-30-22	Shipped	Amazon	Amazon.in	Expedited	J0401	J0401-DR-XX	Western Dre	3XL	B09SDXZBGX	1	INR	885	Debit card	JABALPUR	MADHYA PRA	482002	IN	1
10.42.0.211	04-30-22	Shipped - Del	Merchant	Amazon.in	Standard	JNE2133	JNE2133-KR-	kurta	M	B07D4YRB3F	1	INR	424	Credit Card	MUMBAI	MAHARASHT	400092	IN	0

8.SYSTEM TESTING

8.1 INTRODUCTION

TESTING METHODOLOGIES

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**
- **Output Testing.**
- **Validation Testing.**

Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module . Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection . This test focuses on each module individually, ensuring that it functions properly as a unit . Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification . All important processing path are tested for the expected results . All error handling paths are also tested .

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction . After the software has been integrated a set of high order tests are conducted . The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design .

The following are the types of Integration Testing:

1. **Top-Down Integration**

This method is an incremental approach to the construction of program structure . Modules are integrated by moving downward through the control hierarchy, beginning with the main program module . The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner .

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards .

2. **Bottom-up Integration**

This method begins the construction and testing with the modules at the lowest level in the program structure . Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated . The bottom-up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function .
- A driver (i .e .) the control program for testing is written to coordinate test case input and output .
- The cluster is tested .
- Drivers are removed and clusters are combined moving upward in the program structure
- The bottom-up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality .

8.2 UNIT TESTING:

Unit testing is crucial in ensuring the reliability and accuracy of fraud detection models. It involves testing individual components, such as the Support Vector Machine (SVM) classifier, feature extraction process, and process mining model, to verify their correctness in detecting fraudulent transactions.

For this project, unit tests are designed to:

- Validate the classification accuracy of the SVM model.
- Ensure the event log processing correctly extracts user behaviour patterns.
- Check for edge cases where transactions are borderline fraudulent.

8.3 INTEGRATION TESTING:

Integration testing is essential to ensure that different components of the fraud detection system work together seamlessly. It verifies the correct interaction between the Support Vector Machine (SVM) model, feature extraction process, process mining model, and other system components.

For this project, integration tests are designed to:

- Validate the data flow between event log processing and fraud detection models.
- Ensure the fraud detection results are accurately integrated into the reporting system.
- Test the interaction between APIs and the database to confirm correct data retrieval and storage.

8.4 VALIDATION TESTING:

Validation testing ensures that a software product meets user requirements and functions as intended. It is conducted at the final stages of development to confirm that the system performs correctly in real-world scenarios. This process includes system testing, user acceptance testing (UAT), and functional testing to verify that all features work as expected. Ultimately, validation testing ensures that the right product is delivered to users.

For this project, validation testing is designed to:

- Ensure that the fraud detection model correctly identifies fraudulent transactions with high accuracy.
- Validate that input data preprocessing and feature extraction work correctly for all transaction logs.
- Confirm that system performance remains stable under different loads and transaction volumes.
- Check that security measures are in place to protect sensitive transaction data.

8.5 OUTPUT TESTING:

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

For this project, output testing is designed to:

- Validate that the fraud detection model correctly classifies transactions as fraudulent or legitimate.
- Ensure that generated reports display accurate details of detected fraudulent activities.
- Verify that real-time alerts and notifications are triggered correctly for suspicious transactions.

8.6 USER ACCEPTANCE TESTING:

User Acceptance Testing (UAT) is the final phase of testing, ensuring that the fraud detection system meets business and user requirements before deployment. It focuses on validating the system's usability, accuracy, and overall functionality from an end-user perspective.

For this project, UAT is designed to:

- Ensure that the fraud detection model correctly identifies fraudulent and legitimate transactions.
- Validate that the user interface (UI) is intuitive, responsive, and easy to navigate.
- Confirm that real-time fraud alerts and reports are accurately generated and displayed.
- Gather feedback from end users (such as security analysts or administrators) to assess usability and performance.

8 .7 USER TRAINING:

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed . For this purpose the normal working of the project was demonstrated to the prospective users . Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy .

8 .8 MAINTAINANCE:

This covers a wide range of activities including correcting code and design errors . To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development . Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent . With development in technology, it may be possible to add many more features based on the requirements in future . The coding and designing is simple and easy to understand which will make maintenance easier .

8.9 TESTING STRATEGY

A strategy for system testing integrates sysem test cases and design techniques into a well planned series of steps that results in the successful construction of software . The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding . Testing represents an interesting anomaly for the software . Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing .

9. RESULT & OUTPUT SCREEN SHOTS

The user will give input to the project through the frontend and it is displayed by running the above code and the outcome is shown below .

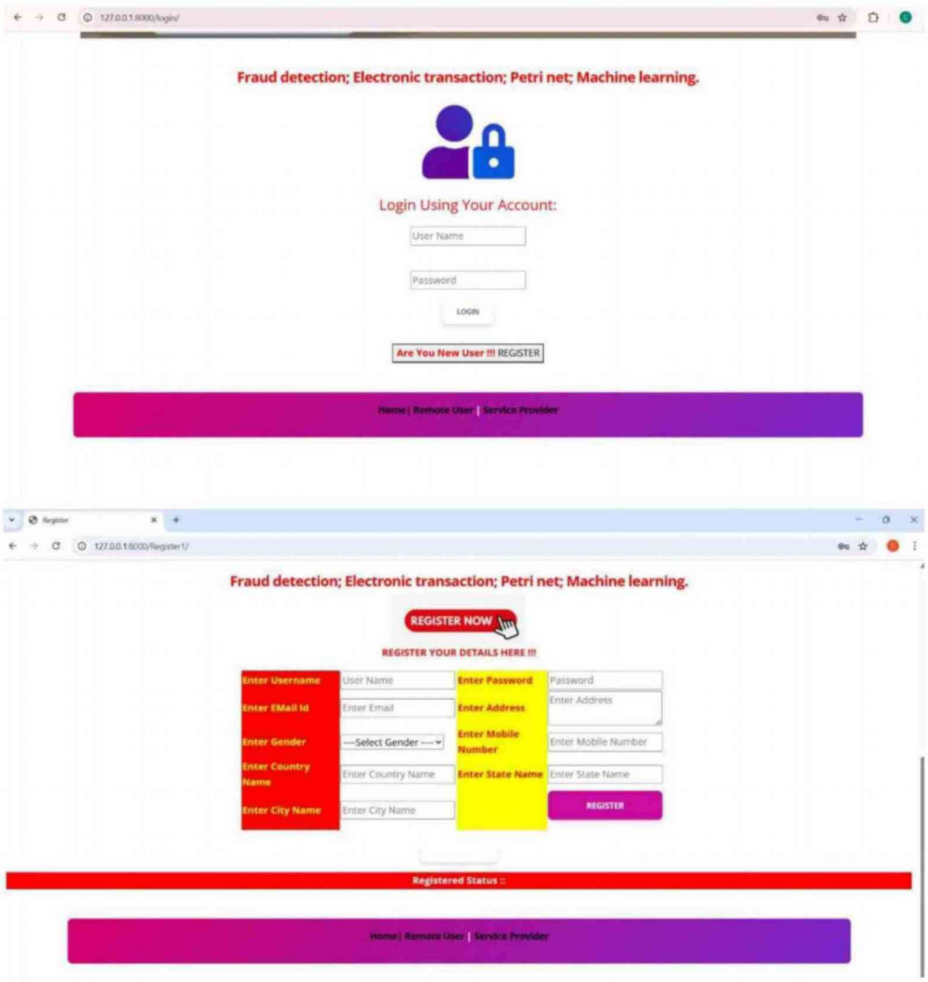


Figure: 9 .1 GRAPHICAL USER INTERFACE

User will enter the order-id, pdate, stats, fulfilment, sales_channel, ship_service_level, style, sku, category, psize, asin, qty, currency values in the boxes so that the values will be calculated in the backend .

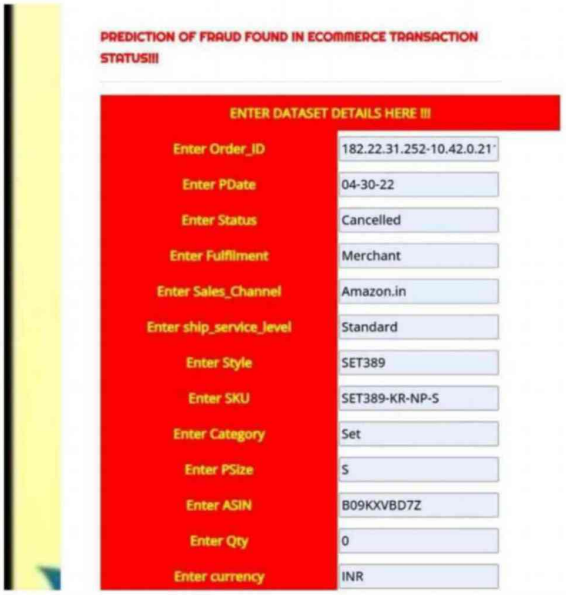


Figure: 9 .2 INPUT FROM USER

Figure: 9 .3 USER DETAILS

After Clicking the Predict button we will go to the prediction page and the output is displayed .

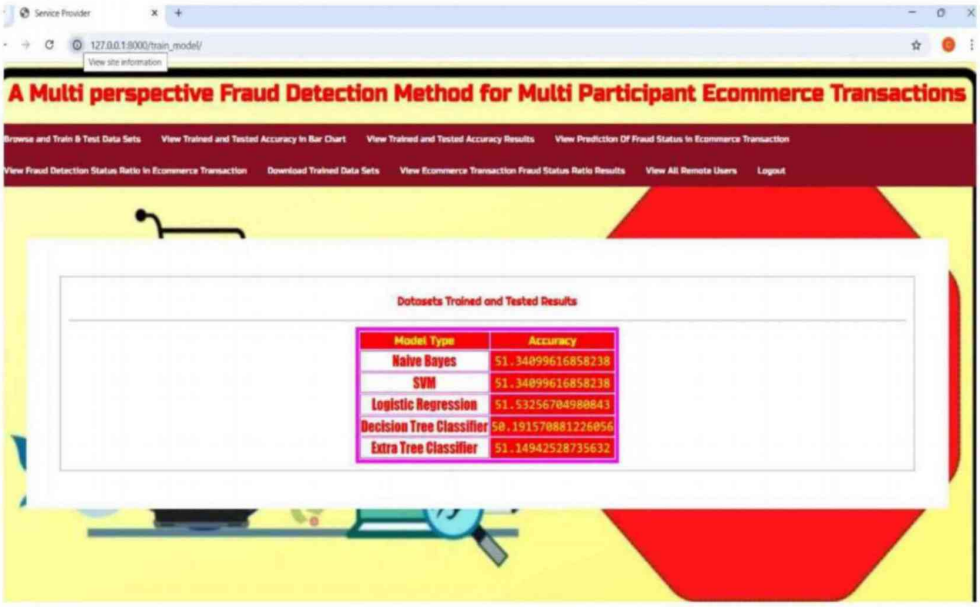


Figure: 9 .4 PREDICTION

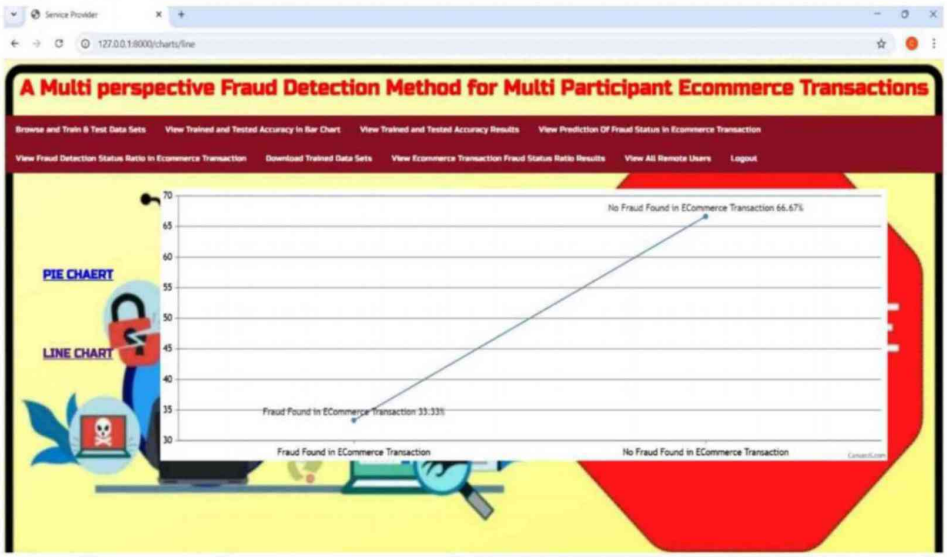


Figure: 9 .5 LINE CHART

10.CONCLUSION & FUTURE ENHANCEMENT

CONCLUSION

This paper proposed a hybrid method to capture fraud transactions by integrating the formal process modeling and the dynamic user behaviors. We analyzed the e-commerce transaction process under five major perspectives: control flow perspective, resource perspective, time perspective, data perspective, and user behavior patterns. This paper utilized high-level Petri nets as the basis of process modeling to model the abnormal user behaviors and created an SVM model to perform fraudulent transaction detection. Our extensive experiments showed that the proposed method can effectively capture fraudulent transactions and behaviors. The overall index of our proposed multi-perspective detection method outperformed the single-perspective detection method. As our future work, related deep learning [38-42] and model checking methods [43-45] would be incorporated in the proposed framework for higher accuracy. Additionally, it's also a future work to incorporate more time features to the behavior patterns so as to make the risk identification more accurate. Furthermore, we will conduct research on constructing a standard fraud mode library, and apply the proposed methodology to other malicious behavior areas by coordinating the models.

FUTURE ENHANCEMENT

Looking ahead, the future of fraud detection in e-commerce transactions lies in the integration of advanced machine learning models with real-time behavioural analytics . The adoption of AI-driven techniques, coupled with process mining methodologies, can enhance the accuracy and efficiency of fraud prevention mechanisms . By leveraging IoT-enabled tracking and decentralized ledger technologies like blockchain, platforms can ensure transparent, immutable transaction records, reducing the risk of fraudulent activities .

Moreover, hybrid fraud detection models that combine the strengths of deep learning, ensemble learning, asnd reinforcement learning can improve anomaly detection capabilities . These models can dynamically adapt to evolving fraud patterns by analysing user behaviour from multiple perspectives, including transaction history, browsing behaviour, and device fingerprints .

Future research can focus on enhancing **spatiotemporal fraud detection**, capturing fraudulent activities across different regions and marketplaces . Leveraging advanced data analytics, network analysis, and real-time monitoring will enable more proactive fraud prevention strategies . The integration of explainable AI (XAI) can further improve model transparency, ensuring that e-commerce platforms can interpret and justify fraud alerts effectively . Continuous innovation in fraud detection methodologies will empower businesses to mitigate risks, safeguard consumer trust, and enhance overall transaction security .

Solutions to Prevent E-commerce Fraud

Addressing fraud in multi-participant e-commerce transactions requires a multi-layered security approach, integrating machine learning models, process mining techniques, and real-time data validation . Below are key measures:

1. Implement AI-driven Behaviour Analysis: AI models can monitor transaction behaviours in real time, identifying unusual spending patterns or login anomalies that indicate fraud .
2. Deploy Multi-factor Authentication (MFA): Enforcing MFA for user logins and transactions can add an additional layer of security, preventing unauthorized access .
3. Leverage Blockchain for Secure Transactions: Decentralized ledger systems ensure transparency and immutability in transaction records, reducing the risk of data tampering .
4. Monitor IP and Device Fingerprints: Tracking user login locations, devices, and IP addresses can help detect suspicious logins from unknown locations .
5. Use Real-time Process Mining: Analysing transactional workflows can help identify abnormal behaviours and deviations from standard purchase processes .
6. Prevent Automated Attacks with CAPTCHA & Bot Detection: Implementing AI-powered CAPTCHA systems can block bot-driven fraudulent activities .
7. Enable Dynamic Risk Scoring: Assigning dynamic risk scores to users and transactions based on behavioural analytics can trigger additional security checks for high-risk transactions .
8. Enforce Digital Identity Verification: Using biometric authentication and document verification can prevent fake or stolen identity-based fraud .
9. Monitor Cross-platform Transactions: Fraudsters often exploit multiple platforms for fraudulent activities . Integrating data analytics across different e-commerce platforms helps in detecting coordinated fraud schemes .
10. Enhance Customer Awareness and Reporting Mechanisms: Educating users about common fraud tactics and enabling them to report suspicious activities can create a safer transaction environment .

Community-led Fraud Prevention Initiative:

Just as environmental sustainability initiatives focus on collective responsibility, fraud prevention in e-commerce requires a community-driven approach . A dedicated network of businesses, consumers, and cybersecurity experts can collaborate to develop fraud intelligence-sharing platforms . These platforms will allow real-time data exchange on emerging fraud tactics, enhancing proactive fraud mitigation strategies .

For example, an initiative could focus on crowdsourcing fraud reports, similar to how environmental groups collect and distribute tree seeds . By engaging users in reporting fraudulent activities, e-commerce platforms can build a fraud awareness ecosystem . This approach strengthens fraud detection models and reinforces trust and security in online transactions .

By continuously innovating and integrating multi-perspective fraud detection mechanisms, e-commerce platforms can create a secure, transparent, and fraud-resilient digital economy .

11.REFERENCES

[1] R . A Kuscu, Y . Cicekcisoy, and U . Bozoklu, *Electronic Payment Systems in Electronic Commerce* . Turkey: IGI Global, 2020, pp . 114- 139 .

[2] M . Abdelrhim, and A . Elsayed, "The Effect of COVID-19 Spread on the e-commerce market: The case of the 5 largest e-commerce companies in the world ." *Available at SSRN 3621166*, 2020, doi: 10 .2139/ssrn .3621166 .

[3] P . Rao et al ., "The e-commerce supply chain and environmental sustainability: An empirical investigation on the online retail sector ." *Cogent . Bus . Manag* ., vol . 8, no . 1, pp . 1938377, 2021 .

[4] S . D . Dhobe, K . K . Tighare, and S . S . Dake, "A review on prevention of fraud in electronic payment gateway using secret code," *Int . J . Res . Eng . Sci . Manag* ., vol . 3, no . 1, pp . 602-606, Jun . 2020 .

[5] A . Abdallah, M . A . Maarof, and A . Zainal, "Fraud detection system: A survey," *J . Netw . Comput . Appl* ., vol . 68, pp . 90-113, Apr . 2016 .

[6] E . A . Minastireanu, and G . Mesnita, "An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection," *Info . Econ* ., vol . 23, no . 1, 2019 .

[7] X . Niu, L . Wang, and X . Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," *arXiv preprint arXiv*: vol . 1904, no . 10604, 2019, doi: 10 .48550/arXiv .1904 .10604 .

[8] L . Zheng et al ., "Transaction Fraud Detection Based on Total Order Relation and Behavior Diversity," *IEEE Trans . Computat . Social Syst* ., vol . 5, no . 3, pp . 796-806, 2018.

[9] Z . Li, G . Liu, and C . Jiang, "Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection," *IEEE Trans . Computat . Social Syst* ., vol . 7, no . 2, pp . 569-579, 2020 .

[10] I . M . Mary, and M . Priyadharsini, "Online Transaction Fraud Detection System," in *2021 Int . Conf . Adv . C . Inno . Tech . Engr* . (*ICACITE*), 2021, pp . 14-16 .

[11] D . Choi, and K . Lee, "Machine learning based approach to financial fraud detection process in mobile payment system," *IT Conv . P* . (*INPRA*), vol . 5, no . 4, pp . 12-24, 2017 .

[12] R . Sarno et al ., "Hybrid Association Rule Learning and Process Mining for Fraud Detection," *IAENG Int . J . C . Sci* ., vol . 42, no . 2, 2015 .

[13] J . J . Stoop, "Process mining and fraud detection-A case study on the theoretical and practical value of using process mining for the detection of fraudulent behavior in the procurement process," M .S . thesis, Netherlands, ENS: University of Twente, 2012 .

[14] M . Jans et al ., "A business process mining application for internal transaction fraud mitigation," *Expert Syst . Appl* ., vol . 38, no . 10, pp . 13351-13359, 2011 .

[15] C . Rinner et al ., "Process mining and conformance checking of long running processes in the context of melanoma surveillance," *Int . J . Env . Res . Pub . He* ., vol . 15, no . 12, pp . 2809, 2018 .

[16] E . Asare, L . Wang, and X . Fang, "Conformance Checking: Workflow of Hospitals and

Workflow of Open-Source EMRs," *IEEE Access*, vol . 8, pp . 139546-139566, 2020 .
[17] W . Chomyat and W . Premchaiswadi, "Process mining on medical treatment history using conformance checking," in *2016 14th Int . Conf . ICT K . Eng . (ICT&KE)*, 2016, pp . 77-83 .
[18] M . D . Leoni, W . M . Van Der Aalst, and B . F . V . Dongen, "Data-and resource-aware conformance checking of business processes," in *Int . Conf . Bus . Info . Sys .*, Springer, Berlin, Heidelberg, 2012 . pp . 48-59 .
S . M . Najem, and S . M . Kadeem, "A survey on fraud detection techniques in ecommerce," *Tech-Knowledge*, vol . 1, no . 1, pp . 33-47, 2021 .
[20] K . Böhmer, and S . Rinderle-Ma, "Anomaly detection in business process runtime behavior--challenges and limitations," *arXiv preprint arXiv*, 2017, doi: 10 .48550/arXiv .1705 .06659 .
[21] K . D . Febriyanti, R . Sarno and Y . Effendi, "Fraud detection on event logs using fuzzy association rule learning," in *2017 11th Int . Conf . Info . Comm . Tech . Sys .*, Surabaya, Indonesia, 2017, pp . 149-154 .
[22] T . Chiu, Y . Wang and M . Vasarhelyi, "A framework of applying process mining for fraud scheme detection," *SSRN Electronic Journal*, 2017, doi:10 .2139/ssrn .2995286 .
[23] W . Yang et al ., "Show Me the Money! Finding Flawed Implementations of Third-party In-app Payment in Android Apps," in *Proc . NDSS*, Shanghai, China, 2017 .
[24] W . Rui, S . Chen, X . Wang and S .Qadeer, "How to Shop for Free Online--Security Analysis of Cashier-as-a-Service Based Web Stores," in *Proc . SSP*,
[25] "Where did I misbehave? Diagnostic information in compliance checking," in *BPM .*, Berlin,
[26] Decomposing alignment-based conformance checking of data-aware process models," in *Proc . OTM*, Amantea, Italy, 2014, pp . 3-20 .
[27] K . Jensen, "Coloured Petri Nets: A High Level Language for System Design and Analysis," *DAIMI Report Series*, vol . 19, no . 338, pp . 342- 416, Mar . 1993 .
[28] B . Ji ., H . Li ., W . Han and Y . Jia, "Research on e-commerce-oriented user abnormal behaviour detection," *Netinfo Security*, Sep . 2014 .
[29] R . Agrawal and R . Srikant, "Fast algorithms for mining association
rules," in *Proc . VLDB*, S . F ., USA, 1994, pp . 487-499 .
[30] R . Srikant and R . Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proc . EDBT*,
Avignon, France, 1996, pp . 1-17 .
[31] Y . Lian, Y . Dai and H . Wang, "Anomaly detection of user behaviors based on profile mining," *Chinese J . Computat-Ch .*, vol . 25, no . 3, pp . 325-330, Mar . 2002 .
[32] C . Cortes and V Vapnik, "Support-vector networks," *Machine Learning*, vol . 20, no . 3, pp .273-297, Sep . 1995.

12.BIBLIOGRAPHY

1. **Fawcett, T., & Provost, F. (1997).** Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3), 291-316.
2. **Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011).** Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), 602-613.
3. **Zhang, Y., & Han, W. (2020).** A hybrid model for fraud detection in e-commerce using machine learning and process mining. *Journal of Information Security and Applications*, 54, 102539.
4. **West, J., Bhattacharya, M., & Islam, R. (2016).** Intelligent financial fraud detection: A comprehensive review. *Computers & Security*, 57, 47-66.
5. **Bauder, R. A., & Khoshgoftaar, T. M. (2018).** A survey of credit card fraud detection using machine learning techniques. *Journal of Big Data*, 5(1), 1-24.
6. **Bolton, R. J., & Hand, D. J. (2002).** Statistical fraud detection: A review. *Statistical Science*, 17(3), 235-255.
7. **Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017).** Credit card fraud detection using machine learning techniques: A comparative analysis. *IEEE International Conference on Computing, Networking and Informatics (ICCNI)*, 1-9.
8. **Cai, H., Zhu, R., & Li, L. (2021).** A real-time fraud detection approach in e-commerce based on graph neural networks. *IEEE Transactions on Computational Social Systems*, 8(2), 257-269.
9. **Phua, C., Lee, V., Smith, K., & Gayler, R. (2010).** A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.
10. **Weinzierl, S., Hommes, S., & Pfitzmann, B. (2012).** Online fraud detection with process mining in e-commerce payment transactions. *International Conference on Business Process Management*, 23-34.

A Multi-perspective Fraud Detection Method for Multi-Participant E-commerce Transactions

WangYang Yu, YaDi Wang, Lu Liu, YiSheng An, Bo Yuan, and John Panneerselvam

***Abstract*—Detection and prevention of fraudulent transactions in e-commerce platforms have always been the focus of transaction security systems. However, due to the concealment of e-commerce, it is not easy to capture attackers solely based on the historic order information. Many researches try to develop technologies to prevent the frauds, which have not considered the dynamic behaviors of users from multiple perspectives. This leads to an inefficient detection of fraudulent behaviors. To this end, this paper proposes a novel fraud detection method that integrates machine-learning and process mining models to monitor real-time user behaviors. First, we establish a process model concerning the B2C e-commerce platform, by incorporating the detection of user behaviors. Second, a method for analyzing abnormalities that can extract important features from event logs is presented. Then, we feed the extracted features to a Support Vector Machine (SVM) based classification model that can detect fraud behaviors. We demonstrate the effectiveness of our method in capturing dynamic fraudulent behaviors in e-commerce systems through the experiments.**

***keywords*—Fraud detection; Electronic transaction; Petri net; Machine learning**

I. Introduction

With the increasing popularity of e-commerce platforms, more and more commercial transactions are now relying on web-based systems than the traditional cash-based approach [1]. Although the entity economy is greatly impacted by the COVID-19 epidemic in recent years, e-commerce remains largely unaffected by the pandemic, whereby aiding a steady market growth [2]. The sales volume of B2C (Business to Customer) e-commerce is expected to reach 6.5 trillion dollars by 2023 [3].

Though the growth of e-commerce and the expansion of modern technologies offer better opportunities for online businesses, new security threats have emerged over the past few years. Reportedly, the significant increase in the number of online fraud cases costs billions of dollars worldwide every year [4]. The dynamic and distributed nature of the Internet has made anti-fraud systems inevitable to ensure the security of online transactions. Existing fraud detection systems focusing on detecting abnormal user behaviors still characterize vulnerabilities when mitigating emerging security threats. An important issue in existing fraud detection systems is their lack of efficient process management during the trading process. The imperfect monitoring function is one of the key issues that need attention [5]. The detection perspective is usually not enough due to the lack of process capture for the existing work. To this end, we propose a process-based method, where user behaviors are recorded and analyzed in real-time, and historical data is transformed into controllable data. In addition, we incorporate a multi-perspective detection of abnormal behaviors.

This paper combines the advantages of process mining and machine learning models by introducing a hybrid method to solve the anomaly detection in data flows, which provides information about each action embedded in a control flow model. By modeling and analyzing the business process of the e-commerce system, this method can dynamically detect changes in user behaviors, transaction processes, and non-compliance situations, and comprehensively analyze and identify fraudulent transactions from multiple perspectives. Important contributions of this paper are listed as follows:

1) A conformance checking method based on process mining is applied in the field of e-commerce transactions to capture the abnormalities.
2) A user behavior detection method is proposed to perform comprehensive anomaly detection based on Petri nets.
3) An SVM model is developed by embedding a multi-perspective process mining into machine learning methods to automatically classify fraudulent behaviors.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 presents a model analysis and a background study. Section 4 forms the theoretical basis and describes our proposed fraud detection method. Section 5 presents and discusses the results of our experiments and Section 6 validates our proposed fraud detection method. Section 7 concludes our paper along with outlining our future research directions.

II. Related Work

Existing fraud detection methods are categorized into non-formal approaches such as machine learning, and formal approaches such as process mining.

The machine-learning-based methods learn from previously obtained historical data to perform classifications

This work is supported in part by the Natural Science Foundation of Shaanxi Province under Grants 2021JM-205, National Natural Science Foundation of China under Grant 52172325, and in part by the fundamental research funds for the central universities under Grant 300102242902. (*Corresponding Authors*: *YaDi Wang and Lu Liu*).

W. Yu and Y. Wang are with the Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China, and also with the School of Computer Science, Shaanxi Normal University, Xi'an 710119, China (E-mail: ywy191@snnu.edu.cn and wyd@snnu.edu.cn).

L. Liu, B Yuan and J Panneerselvam are with the School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, U.K. (E-mail: l.liu@leicester.ac.uk, b.yuan@leicester.ac.uk and j.panneerselvam@leicester.ac.uk).

Y. An is with the School of Information Engineering, Chang'an University, Xi'an, China (E-mail: aysm@chd.edu.cn).

Manuscript received ***, 2022; revised ***, 2022.

or predictions of future observations to identify potential risky offline or online transactions [6]. Xuetong Niu et al. conducted a comparative study on credit card fraud detection methods that rely on machine-learning algorithms. Most of the machine-learning models perform well on the dataset of credit card transactions. Moreover, supervised models perform slightly better than unsupervised models after additional pre-processing, such as removing outliers [7].

Credit card fraud detection is widely deployed at the application layer, which uses the idea of discovering specific abnormal user behaviors to detect fraud. The supervised learning algorithm is the most commonly used learning method in online fraud monitoring transactions, since it has higher accuracy and coverage. Recent research in [8, 9] has proved that the machine learning method can efficiently capture fraudulent transactions in credit card applications. Fraudsters often change their behavioral pattern dynamically to overcome existing fraud detection methods. In online credit card fraud detection, SVM can classify user behaviors under complex scenarios and deliver reliable results [10]. Many researchers take the advantage of combining multiple detection methods for comprehensive fraud detection. For example, focusing on payment fraud applications, Dahee Choi et al. proposed a method by combining supervised and unsupervised learning [11]. Most of the machine learning- based methods use historical data to analyze fraudulent transactions. They have not given enough emphasis to the transactional process flow and dynamic user behaviors.

The second type of fraud detection methods uses process mining, focusing on extracting knowledge from existing event logs in information systems for the purpose of monitoring and improving the operational process in business IT infrastructure [12]. Process mining specializes in comparing the event log with an established model to further detect, locate, and interpret the deviation between the established model and the actual event log [13].

Process mining can detect a large number of abnormal transactions, which are not known to be identifiable by traditional methods. M Jans et al. postulated the emerging process mining approach as an appropriate solution to mitigate against fraud incorporating internal affairs [14]. For example, C Rinner et al. applied conformance checks to monitor the process of melanoma patients [15]. Asare et al. applied alignment and replay to check the conformance of the electronic medical record log and the hospital workflow model [16]. Research has focused on monitoring and evaluating the sequence of processes occurring in the historical medical event log by establishing corresponding training and testing models for conformance checking [17]. Tools such as ProM, Disco and Heustic miner are largely used for conformance checking. Process mining can be an efficient approach for fraud detection.

Especially, it is important to be dynamic and multi-perspective when detecting fraudulent user behaviors [18]. Process mining helps to compare the actual data against the standard model to identify outliers. Despite existing progress in fraud detection, it is still necessary to develop hybrid- learning methods to improve the accuracy of detection [19]. To promote the understanding and development of process mining for anomaly detection, a method of multi-perspective anomaly detection is proposed that goes beyond the perspective of control flow including time and resources [20]. Febriyanti et al. [21] assumed any noticeable changes in business processes as a suspected fraud behavior and proposed a method to detect some suspicious abnormal behaviors using a hybrid method of association rules and process mining. Previous research on using process mining to detect fraudulent transactions showed that process mining is capable of detecting fraudulent transactions, and it can effectively prevent audit fraud at a much earlier stage due to the continuous monitoring nature of event logs [22].

In conclusion, many of the existing machine learning methods only consider static user behaviors based on their occurrence rate. Only a very few studies have investigated real-time, dynamic, and multi-perspective factors of user behaviors in the e-commerce transaction process, which offers great control of the entire transaction process. The detection system based on process mining can record and analyze the changes in user behaviors and their preferences on time. However, analysis of complex details increases the number of variables or factors that should be considered, which makes the detection model more complex.

III. Model Analysis

An e-commerce platform is an information interaction platform that provides online transactions for enterprises and/or individuals. The coverage rate of B2C (Business to Customer) e-commerce platforms is higher than that of other e-commerce platforms, and B2C has become the mainstream model of e-commerce in China [23]. The recent market trend of e-commerce has given emergence to various types of electronic payment systems. Third-party payment platforms supervise and restrict both the buyers and merchants within the terms of the transaction, thereby ensuring the legitimate rights and interests of both buyers and sellers. The process of e-commerce transactions is abstracted and the process flow is established as follow.

A. Process analysis

In a typical B2C process, buyers, e-commerce platforms, and sellers interact with each other. As shown in Fig. 1, the electronic transaction process encompasses five different participants including *Seller*, *Buyer*, the third-party cashier *TP*, the B2C trading platform *BCS* (Buyer and Seller Server), and the cashier server *CS*. This paper summarizes the transaction payment process as follows:

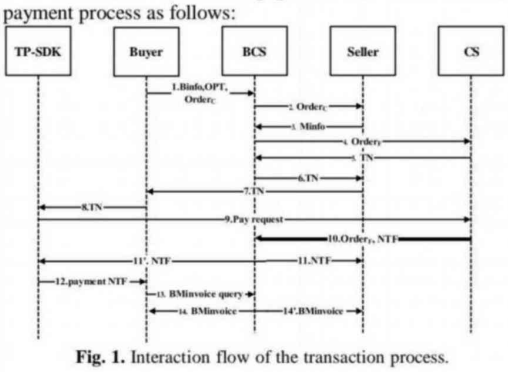


Fig. 1. Interaction flow of the transaction process.

1) After the *Buyer* logs in, the *Buyer* performs a series of operations on the user client device to purchase goods or services. The *BCS* generates a commodity order $order_C$ according to the products or services that are purchased. Commodity order $order_C$ is then passed to the Seller through the e-commerce platform.
2) The Seller makes a decision based on the information in $order_C$, and the Seller passes on the willingness to the platform *BCS*. If the order is rejected, the *Buyer* returns to the user operation process; if the order is accepted, the system establishes a pre-payment formal order $order_F$, which contains the detailed information purchase of the user.
3) After the payment is completed, *CS* signs the formal order $order_F$. Then *CS* sends two payment completion notifications *NTF*, of which one is to notify *TP*, and the other is to notify the seller. The buyer's click triggers the *UpdateOrderStatus* function, in other words, the order status is updated. Afterward, the paid order information $order_F$' is generated.
4) The *Seller* checks the order invoice with the payment status. After that, the *BCS* checks the order and current transaction details.
5) To notify the *CS* of the upcoming payment, *CS* generates a unique transaction number (TN) of the payment information, and then *CS* passes this transaction code to the platform server *BCS*.
6) After the *Seller* receives the TN, it signs and passes TN to the buyer client. At this time, the *Buyer* can confirm the order payment information and enters the password, or cancels the order. Then, the *Buyer* requests payment and enters a password. If the password is correct, the process proceeds to the next step. Otherwise, the transaction fails.
7) The third-party payment client *TP* processes the request, and verifies the credit score and signature of the user. If it is normal, the *TP* makes the payment and sends the payment request command to the *CS*.

B. Fraud mode analysis

To capture the fraudulent behaviors effectively, we define some common fraud modes [23][24] and abstract them as follows.

1) Fraud mode one - an order is tempered by a malicious actor:
   The malicious actor may deceive the victim merchant by sending a fake formal payment order $order_F^A$ to the cashier server. The malicious actor obtained the order items that do not match the payment value by tampering with the order information, such as the total amount.
2) Fraud mode two - subcontract the order:
   The victim pays the malicious actor's order instead of his order. To achieve their goals, the malicious actors impersonate the duties of sellers and buyers. The order information changes before and after the payment.
3) Fraud mode three - send fake notifications:
   During such attacks, the malicious actor submits an order instead of paying for the order, but sends a fake payment result notification to notify the seller that the order is successfully paid.
4) Fraud mode four - paying a cheap order to get expensive goods:
   First, the malicious actor submits a cheap order as an ordinary buyer, and then submits an expensive order but does not pay. However, the system marks the order as "pending". The malicious actor replaces the paid order with the current order at this time.

IV. ANALYTICAL METHODS

Fig. 2 depicts the framework of our detection method. Firstly, the transaction event log is filtered and cleaned, and a database of user behavior mode is constructed in the data preparation stage. Secondly, we perform an analysis on the control flow, resource, throughput time analysis, data flow, and user behavior on the event logs, and extract the abnormalities of each transition from different perspectives as the training features of fraud detection. Then machine- learning algorithms are implemented, and finally, an SVM model is built to classify fraudulent transactions. Our proposed fraud detection method is introduced in detail as follows.

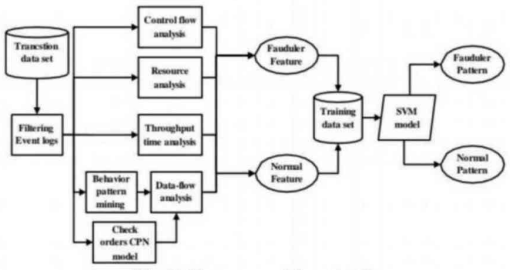


Fig. 2. The proposed framework.

A. Theoretical basis

An event log is made of multiple traces. Each trace represents the life cycle of one case [25], which is specifically composed of case, event, timestamp, action, and resource.

This section introduces establishing the link between the current action, which is shown as an event log, and the action of the model. When some real-life event log is replayed on the process model, some transitions are introduced for routing purposes rather than representing the actual work [26]. We only consider actions of practical significance using a Labeled Petri net defined as follows.

Definition 1. System Net [26]

SN is a system net $SN=(lPN, M_{init}, M_{final})$, $lPN=(P, T, F, l)$ is a Petri net with a labeling function, U_A is defined as universe of action labels, then the label can be formally defined as: $l=T \nrightarrow U_A$; M_{init} is the initial markings and M_{final} is the final markings, which are the tokens contained within the markings of the Petri net.

Definition 2. Data Petri net $DPN=(SN, V, U, R, W, G)$ in which:

- *SN* is a system net $SN=(lPN, M_{init}, M_{final})$ based on $lPN=(P, T, F, l)$;
- V is a set of data variables that are used in the transitions;
- U is a function that defines the range of each value, i.e., D_v is the domain of variable values v, and the value of all variables must be within the range defined. For each value $v \in V$, $U(v)=D_v$;
- R is a read function $R \in T \rightarrow \rho(V)$, which indicates the sets of variables that should be read for each transition;
- W is a write function $W \in T \rightarrow \rho(W)$, which indicates the sets of variables that should be written for each transition;
- G is a guard function $G \in T \rightarrow (V_W \cup V_R)$, which is represented by some combination rules of reading variables and writing variables such that for any transition $t \in T$, and for any variables $v \in V$, if v_r in $G(t)$, then $v \in R(t)$, for any variables $v \in V$, if v_w in $G(t)$, then $v \in W(t)$;

$(DPN, (M, s))[b> (M', s')$ describes an enabled binding b in marking (M, s) may occur. The result is the marking (M', s') after the occurrence. It represents the transition of a net system from one state to another. In *DPN*, the new transitions after triggering should update the newly written variables to all variable sets, i.e., $\mathbb{V}_{new} = \mathbb{V} \oplus \mathbb{V}$, in where $s_{new}(v) = w(v)$ for all $v \in write(t)$, and $s_{new}(v) = s(v)$ for all $v \in V \backslash write(t)$.

Definition 3. Trace and event logs [26]

U_{VN} is a universe of variable names, U_{VV} is the universe of variable values, and U_{VM} is the partial mapping from variable names to values, i.e., $U_{VM} = U_{VN} \nrightarrow U_{VV}$;

A trace, which is defined as a set of action sequences with input and output data, can be represented as $\delta \in (U_A \times U_{VM} \times U_{VM})^*$. In the same way, an event log is composed of multiple sets of traces, which can be expressed as $L \in \mathcal{B}((U_A \times U_{VM} \times U_{VM})^*)$.

Definition 4. Cost function with optimal alignment

s_M is the Data Petri net model, and s_L is the event log; γ is defined as the alignment result of s_M and s_L. In order to quantify the degree of deviation, a cost function is used to define the movements that exist in the above alignment results, i.e., $\kappa \in \Sigma \rightarrow R + 0$. For $6(s_L, s_M) \in \Sigma$, if s_L=>> or s_M = >>, then $\mathbb{V}^{std}_{(s_L,s_M)} = 1$; otherwise, $\mathbb{V}^{std}_{(s_L,s_M)} = 0$. The sequence cost is the sum of costs of individual moves in the sequence, i.e., $\mathbb{V}(\gamma) = \sum_{(s_L,s_M)\in\gamma} n(\mathbb{V}_L, \mathbb{V}_M)$. For all alignment results γ' of the event log and Data Petri net model, there is an optimal alignment $\mathbb{V}(\gamma) \leq \mathbb{V}(\gamma')$.

B. Multi-perspective conformance checking

After the rules are formally defined, conformance checking is used to detect abnormalities. Conformance checking requires an alignment of event log L and process model *DPN*, which is the alignment of each single trace $\delta \in L$ and the process model *DPN*.

The event log of the system records detailed information such as the occurrence time, executor, and interaction data in each action. Through conformance checking, some special trajectories that do not match the trajectories of commonly occurring actions are identified, so that anomalies can be initially detected in a single perspective. For some special cases, such as malicious actors fraudulently using legal accounts to conduct illegal operations or even fraudulent actions, comprehensive analysis and judgment should be carried out in combination with the inspection results from multiple perspectives. In this paper, any trace in the event log that does not conform to the model is suspected for potential anomalies, and the following definition is adopted from [26].

Definition 5. Deviations between the event log and process model

A set (*act*, *r*, *w*, *res*, *time*) is defined, where the read variable of the action *act* is *r*, the write variable is *w*, its resource attribute is represented by *res*, and the throughput time is represented by *time*. The traces in event logs are represented as $S_L=U_A \times U_{VM} \times U_{VM}$, and traces of Data Petri net can be represented as $S_{DPN}=T \times U_{VM} \times U_{VM}$. According to the definition in [26], ">>" means that there is no corresponding move. We use this definition to indicate occurrences of deviations. To replay the event log in the model, different types of deviations are defined as follows:

- Deviation only in log: $\{s_L=(l(t), r, w, res, time) \in S_L\} \cap \{s_M =>>\}$;
- Deviation only in *DPN* model: $\{s_M \in S_{DPN}\} \cap \{s_L=>>\}$;
- Deviation in both model and logs with correct data attributes: $\{s_M=(t, r, w) \in S_{DPN}\} \cap \{s_L=(l(t), r, w)\}$;
- Deviation in both model and logs with incorrect data attributes: $\{s_M=(t, r, w) \in S_{DPN}\} \cap \{s_L=(l(t), r_l, w_l)\} \cap \{s_L=(r \neq r_l \mid w \neq w_l)\}$;
- Deviation in resource: $s_L(res) \neq s_M(res)$;
- Deviation in time: $s_L(time)$ = unqualified;
- All other deviations are considered as abnormal.

The identification of unqualified traces is valuable [25]. The focus of our analysis is to obtain a specific meaning of the points that do not conform to the guards, and information that is hidden in the abnormal points. For multiple control-flow alignments, the optimal alignments γ is selected. Fig. 3 shows a Petri model mined from a set of event log. Four deviations exist between traces in the event log and traces in the model, which are represented as grey areas in Table I. According to the path of Petri net model, from the perspective of control flow, the event log, t_0 has occurred twice. The only deviation in the event log means redundant actions. After the occurrence of t_3, there is t_1 rather than t_0, therefore the only deviation in the model representing some actions is skipped. The throughput time of action t_1 in the 5th line does not meet the threshold requirement. Action t_0 has a deviation in the resource, presented in the 6th line of Table I.

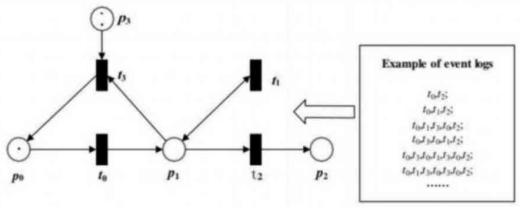


Fig. 3. The deviations example

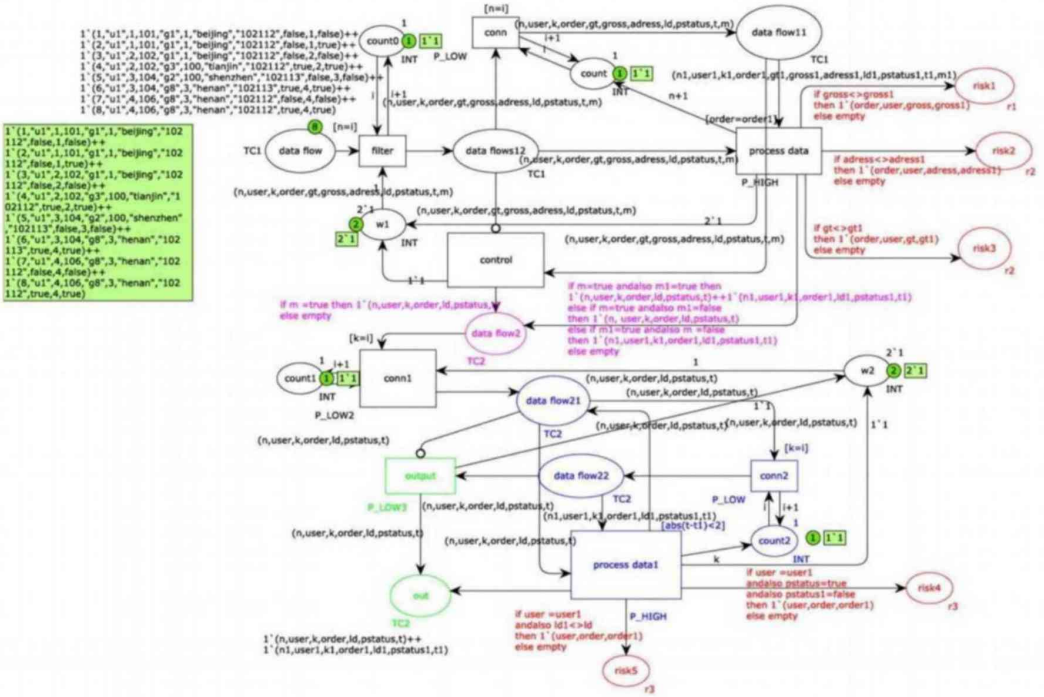


Fig. 4. CPN model: check and process orders.

TABLE I
DEVIATIONS EXAMPLE

	Event log traces	*Model traces*
1	(t_0, {(*att1*:3%), (*att2*:3000)}, {Ø}, {*resource*: Mike}, {*throughput-time*: qualified})	(t_0, {(*att1*:4%), (*att2*:3000)}, {Ø}, {*resource*: Michael}, {*throughput-time*: qualified})
2	(t_0, {(att1:4%), (att2:2000)}, {Ø},{resource:Michael}, {throughput-time: qualified})	>>
3	(t_3, {(*att3*: true)}, {Ø}, {*resource*: Mike}, {*throughput-time*: qualified)	(t_3, {(*att3*: true)}, {Ø}, {*resource*: Mike},{*throughput-time*: qualified)
4	>>	(t_0, {(*att1*:4%), (*att2*:3000)}, {Ø}, {*resource*: Michael}, {*throughput-time*: qualified})
5	(t_1, {(*att4*: VIP)}, {Ø}, {*resource*: Kris}, {*throughput-time*: unqualified)	(t_1, {(*att4*: VIP)}, {Ø}, {*resource*: Kris},{throughput-time: qualified)
6	(t_2, {(*att5*:3)}, {Ø}, {*resource*: Mike}, {*throughput-time*: qualified)	(t_2, {(*att5*:3)}, {Ø}, {*resource*: Amber},{*throughput-time*: qualified)

C. Check and process orders

Colored Petri net (CPN) [27] is a powerful modeling tool for concurrent and distributed systems, which can not only be used to process and analyze users' transaction orders, but to realize the formalization and visualization of the detection process dynamically. The CPN model in Fig. 4 corresponds to the actions *Check and Process Orders* in the business process model. The CPN model is established for detecting and processing order information according to the detection target of actions. The detection target of the CPN model mainly includes: (1) detecting whether there are other unpaid orders under the same user ID in a short period of time; (2) detecting whether the IP address of the order under the same user ID has changed in a short period of time; (3) detecting whether the amount in commodity orders is consistent with the actual payment amount of the final order for the same order from the same user; (4) detecting whether the address in commodity orders is consistent with the actual address of the final order; and (5) detecting whether the goods type in commodity orders is consistent with the actual goods type of the final order.

As shown in Fig. 4, the place "*data flow*11" and the place "*data flow*12" in the CPN model realize the comparison of commodity orders and final transaction orders of the same transaction one by one. The place "*w*1" controls the number and order of detection. The transition "*process data*" can compare the commodity order with the final one. The places "*process data*" aims to find the orders with unusual order information, when its rules set by the arc function are satisfied, the token is the input to the corresponding places "*risk*1", "*risk*2", and "*risk*3" respectively. Among them, the existence of the token in the place "*risk*1" represents the abnormal change in the order amount. These tokens represent order information, such that abnormal orders can be visually observed and extracted.

Before detecting whether a given user has other unpaid orders, the order information flow should be filtered first. This is because the object of this type of anomaly detection is the final orders of the transaction, and our input data flow contains two types of order information. As shown in Fig. 4, the pink part of the model shows the filtering function used to obtain the final transaction orders. Next, the transition

"*process data*1" processes the order data. If the arc function of "*risk*4" is satisfied, a new token is generated in the place "*risk*4"; in the same way, if the arc function pointing to "*risk*5" is satisfied, i.e., a new token is generated in the place "*risk*5". Abnormal order information can now be extracted from the tokens. In summary, the CPN model realizes the functions of processing and detecting order information.

D. User operation behavior detection

Next, we add the data flow perspective based on the above detection method, which integrates the function of user behavior detection. Buyer behavior analysis can be divided into two parts: static attribute and dynamic behavior [28].

The user's static attribute data used in this paper mainly includes IP address, login time, and operation duration. We use the Apriori algorithm [29] to obtain the normal patterns based on the user's historical static attribute data. Before using operational data for mode mining, static attributes should be pre-processed and described using mathematical models. The login time is expressed as an integer of [0, 24]. Table II shows the characteristics of static attributes.

TABLE II
USER STATIC ATTRIBUTE PRE-PROCESSING

Attribute	*IP address*	*Login time*	*Operation duration (min)*
Example	192.168.1.249	21	60

When the user operates on the APP or web browser, a series of operation data is generated; the user's behavior habits are hidden in these data. When other users use the same device, account, and IP address as the actual user, the behavioral patterns obtained from the user's historical behavioral data are used for pattern matching. As shown in Table III, the user behavior data used in this method mainly includes searching for products, browsing products, favorites, adding to shopping carts, viewing shopping carts, and so on. The categories of behavior data are limited, so that integers are used to label and classify user behaviors.

TABLE III
USER BEHAVIOR CATEGORIES AND MATHEMATICAL IDENTIFICATION

Behavior category	*Symbol*
Search	0
Browser	1
Favorite	2
Add to the cart	3
View the cart	4
View the favorites	5
Submit the orders	6

By identifying the temporal logic relationship rules through data mining algorithms, user behavior patterns are obtained. This paper uses the GSP algorithm [30] to mine user behavior data.

Fig. 5 shows the specific steps of user anomaly detection. Pattern matching of the user's operation behavior corresponds to the functions in transition D and U, which are used to detect and analyze the operation and static attributes. (*act*, *att*, *att'*, *alg*) represents the input variable of action *act* is *att*, and the new variable obtained by the algorithm *alg* is *att'*. For example, a set of bindings for dynamic behavior habits can be represented as (*D*, *OPT_type*, *Operation*, *recursive_correlation_ function*). The similarity of action *D* is obtained after processing using the recursive correlation function [31]. A predefined threshold is used to determine whether the current user behavior is abnormal or not. Similarly, a set of valid bindings for static attributes can be represented as (*U*, *Static_User_Pattern*, *Static_att*, *Full_sequence_comparison*). It indicates that the full sequence comparison method [31] is used to compare the current static attribute with the static attribute pattern. Combined with the process, the matching similarity *Static_att* is used as the *w* of the action, and the judgment can be made according to the threshold value.

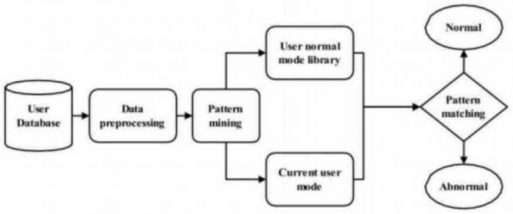


Fig. 5. User data pattern mining and detection process.

E. An anomaly extraction algorithm for e-commerce business process based on multi-perspective

Through the analysis and modeling of the e-commerce transaction process structure mentioned above, the reference model of the e-commerce business process based on Petri net is obtained. The inputs of Algorithm 1 include the e- commerce business process model, the CPN model, and the sequence of actions that occur in real-time. Firstly, the event log and the reference model are optimally aligned. Then according to steps 2.1, 2.2, 2.3, and 2.4, the anomalies of the current transaction's event trajectory in the control flow perspective, resource flow perspective, time flow perspective, and data flow perspective are calculated respectively, and the abnormal point sequence $B_d[T_x]$ corresponding to the abnormal point sequence $B_d[T_x]$ is recorded as 1. The abnormal point sequence $B_d[T_x]$ corresponding to the transition T_x without offset is recorded as 0. Step 3 counts the results of processing orders; Steps 4 and 5 count the results of user static attributes and dynamic behavior detection, and record them in the corresponding abnormal point sequence $B_d[T_x]$. The final algorithm output result is the abnormal point set $B[T_x]$, which is used as the initial data of SVM model training for fraud detection.

Algorithm 1: The anomaly extraction algorithm for e-commerce transaction process based on multi-perspectives

Input:
Business process model S_{DPN}= (*P*, *T*, *F*, *V*, *U*, *R*, *W*, *G*); CPN model of check and process orders; Event logs S_L.

Output:
The sequence of anomalies for SL is stored in B[t_x], $t_x \in T$, x is the serial number, x∈[0, 32].

//Step1 :Initialize parameters.

1 $B=\varnothing$, where $B[t_x]=B_c[t_x]\cup B_r[t_x]\cup B_t[t_x]\cup B_d[t_x]$ respectively represent the set of anomalies of control flow, resource flow, time flow, and data flow that may occur in transition t_x, and the judgment result of user behavior is placed in the set $B_d[t_x]$.

//Step2 : According to the process mining conformance checking algorithm, the optimal alignment result is denoted as γ, which

can be judged as follows.
if $\{s_L=(l(t_x), r, w, res, time)\in S_L\} \cap \{s_M=>>\}$ or $\{s_M\in S_{DPN}\} \cap \{s_L=>>\}$, **then**
  $B_c[t_x]=1$
Else
  $B_c[t_x]=0$
end if
if $s_L(res)$!= $s_M(res)$, **then**
  $B_r[t_x]=1$
Else
  $B_r[t_x]=0$
end if
if $s_L(time)$ = *unqualified*, **then**
  $B_t[t_x]=1$
Else
  $B_t[t_x]=0$
end if
if $\{\{s_M=(t_x, r, w)\in S_{DPN}\} \cap \{s_L=(l(t_x), r_l, w_l)\in S_L\} \cap \{r\neq r_l \mid w\neq w_l\}\} \cup \{s_L(\text{Guards}(t_x)) = False\}$, **then**
  $B_d[t_x]=1$
Else
  $B_d[t_x]=0$
end if

// Step3: For the detection result of order information, the function of CPN model corresponds to transition t_{28} in *DPN* model, the event log S_L is input into the model to obtain the result.
if there exists *token* in the end places $risk_i$, $i\in[1,5]$, **then**
  the abnormal conditions corresponding to $risk_i$ of the end places with *token* are recorded in the sequence $B_d[t_x]$:
  $risk_i \rightarrow B_d[t_{28}]=1$
Else
  $risk_i \rightarrow B_d[t_{28}]=0$
end if
if the static attribute of the user does not meet the threshold, **then**
  $B_d[t_{29}]=1$
Else
  $B_d[t_{29}]=0$
end if
if the user dynamic behavior does not meet the threshold, **then**
  $B_d[t_4]=1$
Else
  $B_d[t_4]=0$
end if
Return $B[t_x]$

F. Fraud detection based on SVM

The evaluation of a single perspective is relatively one- sided and cannot accurately determine whether the current transaction is fraudulent or not. Therefore, it is very important to integrate the detection results of each perspective to evaluate the transaction's status as a whole. Next, the multi- perspective detection results are used as the features, and SVM is used to learn from these features and to integrate them for evaluating whether the current transaction has fraudulent behavior or not as a whole.

1) Classification problem and SVM [32]

The problem of fraud detection is essentially a binary classification problem, which can be solved by a classification model. The binary classification problem is a process in which a classification function judges whether the input data belongs to the positive class or the negative class. The mathematical definition is as follows:

$$h(x) = p(y=1|x), y=0 \text{ or } 1 \quad (1)$$

where, x presents the input data; y presents the class of the input data; $h(x)$ is the classification function.

Anomaly detection is a process in which a detection model uses user data to judge whether the user is abnormal. Anomaly detection satisfies the definition of the classification problem. An SVM model is a supervised learning method that can be used to solve binary classification problems. Compared with other classification methods, SVM delivers better performance with less sample size. In addition, SVM is good at using the kernel function to solve the case where the data is linearly inseparable.

The key role of SVM is to find a suitable hyperplane to divide samples into two classes, and maximize the distance between the samples and the hyperplane. The loss function of SVM is as follows:

$$loss=\sum_{i=1}^{N}\max\left(0,1-y_i\left(\omega^T x_i+b\right)\right)+\lambda\|\omega\|^2 \quad (2)$$

where, x_i is the feature vector of the i-th sample; y_i is the label of the i-th sample; ω is the weight parameter; b is the bias parameter; λ is the regularization coefficient.

Through learning from the dataset and updating the weights, the loss function of the SVM model gradually decreases and finally converges. After the above process, the SVM model is successfully constructed and used for prediction. By taking the features of the current user as input, the SVM model classifies whether the current transaction behaviors are fraudulent or not.

2) Feature selection for anomaly detection

In the process of multi-perspective detection, each perspective gives an inference about whether the current transaction has fraudulent behaviors or not. The SVM model takes the detection inference of these perspectives as features. We obtain 82 anomaly detection features from the Data Petri net and data mining process. These features are used to detect whether a current transaction is abnormal from multiple perspectives. These features are used as the feature vectors in the SVM model. Parts of features and their meanings are shown in Table IV. These features are respectively the control flow analysis results of 20 actions, the time flow analysis results of 20 actions, the resource flow analysis results of 20 actions, and the data flow analysis results of 22 actions.

TABLE IV
EXAMPLES OF FEATURES

Feature	*Feature name*	*Meaning*
X_1	Control flow analysis result of transition A	Whether the control flow of transition A is abnormal
X_2	Control flow analysis result of transition B	Whether the control flow of transition B is abnormal
X_{27}	Time flow analysis result of transition H	Whether the time flow of transition H is abnormal
X_{45}	Resource analysis result of transition F	Whether the resource flow of transition F is abnormal
X_{62}	Data flow analysis result of transition D	Whether the user's operation behavior is abnormal
X_{82}	Data flow analysis result of transition U	Whether the user's static attributes are abnormal

V. PROCESS MINING EXPERIMENT RESULTS

This paper uses the process-mining tool ProM LIte 1.2 as the experimental platform [33]. Data flow experiments and

A. Control flow analysis

According to our proposed method introduced in the previous section, we generate the control flow, as shown in Fig. 6.

Fig. 6. Part of control flow analysis results.

This section uses the plug-in *Replay a Log on Petri Net for Conformance Analysis* to complete the control flow detection. In Fig. 6, we intercepted several traces in the result. The green part represents "move both on log and model", which is normal. The grey part means "move on the model only" which depicts that the event log has no deviations corresponding to the model. The purple part means "move on a log only", which means that the model has no deviation corresponding to the event log, further indicating that the event log is abnormal, that is, skipped actions. For example, actions 5 and 6 are skipped in trace66. This always means that the order placement represented in this trace is not approved by the merchant.

B. Throughput time analysis

Throughput time is the interval among actions, which can be obtained by analyzing the completion time among actions recorded in the event log. We use the plug-in Replay a Log for Performance/Conformance Analysis for the throughput time analysis [34]. Fig. 7 depicts the time interval among each action. The time interval of each action in the actual e-commerce process is very close, and the time difference is in the order of microseconds. For the sake of intuition, this paper extends the running time of each action. We set the lowest threshold of the transition processing time to 10 milliseconds, and the highest threshold to 60 seconds.

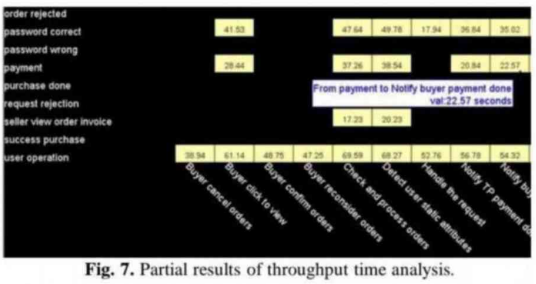


Fig. 7. Partial results of throughput time analysis.

C. Resource analysis

This section analyzes each participant, who performed the actions recorded in the event log. We used the plug-in Multi Perspective Explorer [35] from ProM to detect the outliers in the resource.

In Fig. 8, different actions are completed by different performers and visually displayed in different colors in the analysis panel of ProM. For example, light blue represents the Buyer, and dark blue means that the executor is BCS. The purple flag means that an unauthorized participant has performed the corresponding action. For example, for the action Order Created in the first trace, its executor is an unauthorized user.

D. Data flow analysis

Data flow analysis is introduced to address the shortcomings of control flow analysis. It provides information about each action embedded in the process model [18]. To get the results of data flow analysis, firstly we use the plug-in *Edit Petri Net with Data* to obtain the Data Petri net model. The complete results are shown in Fig. 9.

The guard is the normal threshold set for each action. Specifically, it is normal when the current input value meets the standards set by the guards of an action. When no guard meets the configurations of data flow, that action point is deemed as an abnormal data flow. Action Submit Orders sequence writes the ID and types of the product that the buyer wants to purchase. Fig. 10 is a snippet of this sequence. It can be seen that the input of the action *Order Addressed By Seller* is the goodsID and goodstype.

To arrive at a clearer result, the plug-in Multi-Perspective Process Explorer and Conformance Checking are used to match and analyze the event log and the *DPN*. The result is shown in Fig. 11, where each action is represented with different colors. For instance, green represents the move both on model and log, purple means move on the model only, and grey represents invisible actions, that is, skipped actions. By clicking on a given action, we can obtain the matching information between the model and the event log in the data flow of each action. The data marked in red indicates a mismatch. We extract these suspicious anomalies and use them as the basis for subsequent training using machine- learning models.

VI. EXPERIMENT AND ANALYSIS BASED ON SVM MODEL

This section utilizes the user anomaly detection features as data sources, which are obtained from multi-perspective detection and uses the SVM model to determine whether there exists any fraud. This experiment utilizes the grid search method [36] to adjust the hyper-parameters of the SVM model, the obtained hyper-parameters and the split ratio are chosen to perform a cross-validation experiment.

A. Data-set construction

Each data consists of 82 anomaly detection features, and each feature characterizes a value of 0 or 1, where 1 represents abnormality and 0 represents normality. A representation of the dataset used in our experiments is shown in Table V.

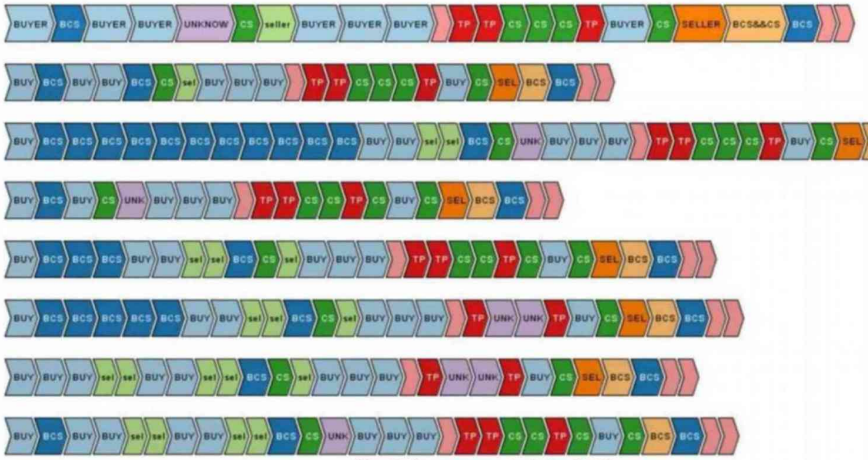


Fig. 8. Resource analysis by ProM.

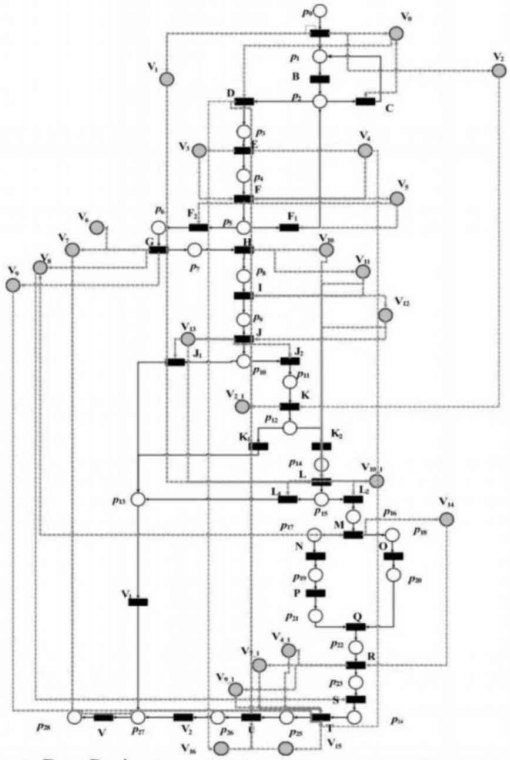


Fig. 9. Data Petri net model of the e-commerce transaction process.

According to Algorithm 1, the training data represented in Table V is obtained, the SVM model is trained by using the obtained dataset, and finally, an anomaly detection model is obtained. For example, in the first row of Table V, each control flow appears to be normal from every perspective, which is marked as 0, and the classification result is 0. In the second row, X_5 marked as 1 represents the action Order Addressed by Seller is skipped, X_{27} marked as 1 represents the execution time of action TN Created by Order takes too long, X_{45} marked as 1 represents the abnormal performer of the action Order Created, and X_{76} marked as 1 means the order payment amount is modified. Herein, this entire sequence is determined as the first fraud mode. In the third row, X_{11} and X_{12} marked as 1 indicate that the payment- related actions are skipped, X_{53} and X_{54} marked as 1 indicate that the performer of the notification action after the payment is abnormal, and X_{68} marked as 1 means that the user's credit level is low and does not meet the threshold. Thus, the entire sequence is determined as the third fraud mode.

TABLE V
EXAMPLES OF TRAINING DATA IN THE DATASET

Example1	*Marking*	*Example2*	*Marking*	*Example3*	*Marking*
X_1	0	…	0	…	0
X_2	0	X_5	1	X_{11}	1
…	0	…	0	X_{12}	1
X_{21}	0	X_{27}	1	…	0
X_{22}	0	…	0	X_{53}	1
X_{23}	0	X_{45}	1	X_{54}	1
X_{24}	0	…	0	…	0
…	0	X_{76}	1	X_{68}	1
X_{82}	0	…	0	…	0

B. Model performance test experiment based on K-fold cross validation method

This section verifies the validity of our SVM-based fraud detection model through comparative experiments. According to the data type, the experiment encompasses three scenarios: control flow characteristic data only, data flow characteristic data only, and both control flow characteristic data and data flow characteristic data. These three cases are tested by cross-validation method respectively to obtain the model precision (Precision), recall rate (Recall), F1-Score and AUC (Area under the ROC Curve) under the current data, whereby the model performance is analyzed according to the above indicators.

$$\begin{cases} precision = \dfrac{TP}{TP + FP} \\ recall = \dfrac{TP}{TP + FN} \end{cases} \quad (3)$$

$$\begin{cases} F1 = \dfrac{2 * precision * recall}{precision + recall} \end{cases}$$

TP represents the number of positive samples that are predicted as a positive class; *FP* represents the number of negative samples that are predicted as a positive class; *FN* represents the number of positive samples that are predicted as a negative class.

We use the grid search method to select the best hyper parameters of the SVM model, which is an enumeration method that is used to select the most optimal combination of hyper parameters. Through the grid search method, the optimal SVM hyper-parameters are obtained, as shown in Table VI.

TABLE VI
HYPER-PARAMETER SETTINGS OF THE SVM MODEL

Hyperparameter	*Meaning*	*Setting*
kernel	Kernel Function	Polynomial Kernel
C	Regularization	3
gamma	Kernel coefficient	0.25
degree	Highest degree of Polynomial Kernel	3
tol	Stop criterion	0.1

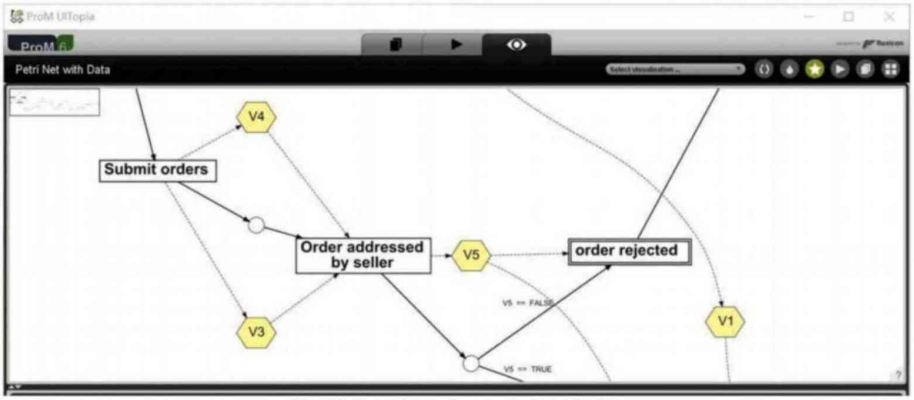


Fig. 10. Part of data flow analysis by ProM.

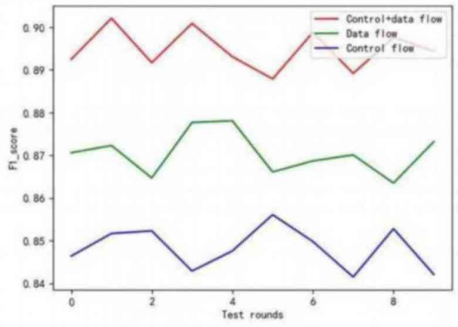
Fig. 11. The result of data flow analysis.

K fold Cross-validation [37] is an effective way of verifying the effectiveness of the model performance. In this experiment, the value of k is 10, that is, the experiment is carried out through 10 fold- cross verification.

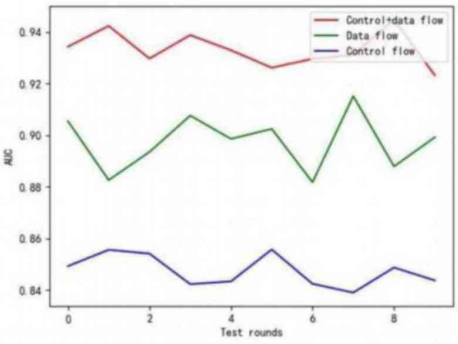
Fig. 12 (a) and (b) represents the statistical detection indicators of F1-core and AUC of our proposed SVM-based fraud detection model, obtained based on the 10-fold cross validation. Among them, the blue curve is the control flow index, the green curve is the model score considering the data flow, and the red curve represents the score under the fusion of multi-perspective features. As seen from Fig. 12, F1-score under the data fusion of control flow and data flow is higher than that when only one type of data is considered, that is, when the data of control flow and data flow are considered comprehensively, better user anomaly detection is obtained.

TABLE VII
FRAUD DETECTION MODEL RESULTS BASED ON SVM

Perspectives	*Precision*	*Recall*	*F1-score*	*AUC*
Control+data flow	0.946	0.852	0.895	0.935
Data flow	0.912	0.837	0.871	0.892
Control flow	0.889	0.812	0.849	0.842



(a) F1-score statistics



(b) AUC statistics

Fig. 12. F1-score and AUC statistics for three situations.

To further validate the fraud detection effects of our model under the three aforementioned cases, we consider various performance indicators under 50 rounds of tests and calculate their average values. The results are shown in Table VII. As seen from Table VII, the index of F1-score and AUC are both greater than indexes that consider only one of the perspectives under the case of integrating data flow and control flow features. These two kinds of characteristic data only consider one aspect of the user's anomaly. After learning the two types of data through the machine-learning model, the information of the two aspects is fully utilized to comprehensively detect user anomalies with better effect.

In summary, when compared with considering only one perspective of information, our proposed model characterizes a higher F1-score and AUC indicators. The detection effect of abnormal e-commerce users is better in our model. Therefore, our proposed method can detect abnormal e-commerce users more comprehensively. In addition, compared with the related deep learning methods for the fraud detection in e- commerce, our methodology can depict the transaction process and structures, and it is interpretable.

VII. CONCLUSION

This paper proposed a hybrid method to capture fraud transactions by integrating the formal process modeling and the dynamic user behaviors. We analyzed the e-commerce transaction process under five major perspectives: control flow perspective, resource perspective, time perspective, data perspective, and user behavior patterns. This paper utilized high-level Petri nets as the basis of process modeling to model the abnormal user behaviors and created an SVM model to perform fraudulent transaction detection. Our extensive experiments showed that the proposed method can effectively capture fraudulent transactions and behaviors. The overall index of our proposed multi-perspective detection method outperformed the single-perspective detection method. As our future work, related deep learning [38-42] and model checking methods [43-45] would be incorporated in the proposed framework for higher accuracy. Additionally, it's also a future work to incorporate more time features to the behavior patterns so as to make the risk identification more accurate. Furthermore, we will conduct research on constructing a standard fraud mode library, and apply the proposed methodology to other malicious behavior areas by coordinating the models.

REFERENCES

[1] R. A. Kuscu, Y. Cicekcisoy, and U. Bozoklu, *Electronic Payment Systems in Electronic Commerce*. Turkey: IGI Global, 2020, pp. 114–139.

[2] M. Abdelrhim, and A. Elsayed, "The Effect of COVID-19 Spread on the e-commerce market: The case of the 5 largest e-commerce companies in the world." *Available at SSRN 3621166*, 2020, doi: 10.2139/ssrn.3621166.

[3] P. Rao et al., "The e-commerce supply chain and environmental sustainability: An empirical investigation on the online retail sector." *Cogent. Bus. Manag.*, vol. 8, no. 1, pp. 1938377, 2021.

[4] S. D. Dhobe, K. K. Tighare, and S. S. Dake, "A review on prevention of fraud in electronic payment gateway using secret code," *Int. J. Res. Eng. Sci. Manag.*, vol. 3, no. 1, pp. 602-606, Jun. 2020.

[5] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90-113, Apr. 2016.

[6] E. A. Minastireanu, and G. Mesnita, "An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection," *Info. Econ.*, vol. 23, no. 1, 2019.

[7] X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," *arXiv preprint arXiv*: vol. 1904, no. 10604, 2019, doi: 10.48550/arXiv.1904.10604.

[8] L. Zheng et al., "Transaction Fraud Detection Based on Total Order Relation and Behavior Diversity," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 3, pp. 796-806, 2018.
[9] Z. Li, G. Liu, and C. Jiang, "Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 569-579, 2020.
[10] I. M. Mary, and M. Priyadharsini, "Online Transaction Fraud Detection System," in *2021 Int. Conf. Adv. C. Inno. Tech. Engr. (ICACITE)*, 2021, pp. 14-16.
[11] D. Choi, and K. Lee, "Machine learning based approach to financial fraud detection process in mobile payment system," *IT Conv. P. (INPRA)*, vol. 5, no. 4, pp. 12-24, 2017.
[12] R. Sarno et al., "Hybrid Association Rule Learning and Process Mining for Fraud Detection," *IAENG Int. J. C. Sci.*, vol. 42, no. 2, 2015.
[13] J. J. Stoop, "Process mining and fraud detection-A case study on the theoretical and practical value of using process mining for the detection of fraudulent behavior in the procurement process," M.S. thesis, Netherlands, ENS: University of Twente, 2012.
[14] M. Jans et al., "A business process mining application for internal transaction fraud mitigation," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13351-13359, 2011.
[15] C. Rinner et al., "Process mining and conformance checking of long running processes in the context of melanoma surveillance," *Int. J. Env. Res. Pub. He.*, vol. 15, no. 12, pp. 2809, 2018.
[16] E. Asare, L. Wang, and X. Fang, "Conformance Checking: Workflow of Hospitals and Workflow of Open-Source EMRs," *IEEE Access*, vol. 8, pp. 139546-139566, 2020.
[17] W. Chomyat and W. Premchaiswadi, "Process mining on medical treatment history using conformance checking," in *2016 14th Int. Conf. ICT K. Eng. (ICT&KE)*, 2016, pp. 77-83.
[18] M. D. Leoni, W. M. Van Der Aalst, and B. F. V. Dongen, "Data-and resource-aware conformance checking of business processes," in *Int. Conf. Bus. Info. Sys.*, Springer, Berlin, Heidelberg, 2012. pp. 48-59.
[19] S. M. Najem, and S. M. Kadeem, "A survey on fraud detection techniques in ecommerce," *Tech-Knowledge*, vol. 1, no. 1, pp. 33-47, 2021.
[20] K. Böhmer, and S. Rinderle-Ma, "Anomaly detection in business process runtime behavior--challenges and limitations," *arXiv preprint arXiv*, 2017, doi: 10.48550/arXiv.1705.06659.
[21] K. D. Febriyanti, R. Sarno and Y. Effendi, "Fraud detection on event logs using fuzzy association rule learning," in *2017 11th Int. Conf. Info. Comm. Tech. Sys.*, Surabaya, Indonesia, 2017, pp. 149-154.
[22] T. Chiu, Y. Wang and M. Vasarhelyi, "A framework of applying process mining for fraud scheme detection," *SSRN Electronic Journal*, 2017, doi:10.2139/ssrn.2995286.
[23] W. Yang et al., "Show Me the Money! Finding Flawed Implementations of Third-party In-app Payment in Android Apps," in *Proc. NDSS*, Shanghai, China, 2017.
[24] W. Rui, S. Chen, X. Wang and S.Qadeer, "How to Shop for Free Online--Security Analysis of Cashier-as-a-Service Based Web Stores," in *Proc. SSP*, Oakland, CA, USA, 2011, pp. 465-480.
[25] E. Ramezani, D. Fahland and W. Aalst, "Where did I misbehave? Diagnostic information in compliance checking," in *BPM.*, Berlin, Germany, Springer, 2012, pp. 262-278.
[26] M. Leoni, J. Munoz-Gama, J. Carmona and W. Aalst, "Decomposing alignment-based conformance checking of data-aware process models," in *Proc. OTM*, Amantea, Italy, 2014, pp. 3-20.
[27] K. Jensen, "Coloured Petri Nets: A High Level Language for System Design and Analysis," *DAIMI Report Series*, vol. 19, no. 338, pp. 342-416, Mar. 1993.
[28] B. Ji., H. Li., W. Han and Y. Jia, "Research on e-commerce-oriented user abnormal behaviour detection," *Netinfo Security*, Sep. 2014.
[29] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. VLDB*, S. F., USA, 1994, pp. 487-499.
[30] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proc. EDBT*, Avignon, France, 1996, pp. 1-17.
[31] Y. Lian, Y. Dai and H. Wang, "Anomaly detection of user behaviors based on profile mining," *Chinese J. Computat-Ch.*, vol. 25, no. 3, pp. 325-330, Mar. 2002.
[32] C. Cortes and V Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp.273-297, Sep. 1995.
[33] F. Yasmin, R. Bemthuis, M. Elhagaly, F. Wijnhoven and F. Bukhsh, "A Process Mining Starting Guideline for Process Analysts and Process Owners: A Practical Process Analytics Guide using ProM," *DSI technical report series*, Jul. 2020.
[34] A. Adriansyah, "Replay a log on petri net for performance/ conformance plug-in," Technische Universiteit Eindhoven, 2012.
[35] F. Mannhardt, M Leoni and H Reijers, "The Multi-perspective Process Explorer," *BPM (Demos)*, vol. 1418, pp. 130-134, Aug. 2015.
[36] D. Chen, X. Liu, Y. Zhou, X. Yang, L. Lu and W. Xin, "Grid search as applied to the determination of Mark-Houwink parameters," *J. Appl. Polym. Sci.*, vol. 76, no. 4, pp. 481-487, 2015.
[37] J. Myerson, L. Green and M. Warusawitharana, "Area under the curve as a measure of discounting," *J. Exp. Anal. Behav.*, vol. 76, no. 2, pp. 235-243, Oct. 2001.
[38] L. Zheng, G Liu, C. Yan, C Jiang and M. Li, "Improved TrAdaBoost and its Application to Transaction Fraud Detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 5, pp. 1304-1316, Jul. 2020.
[39] J. Cui, C Yan and C Wang, "ReMEMBeR: Ranking Metric Embedding-Based Multicontextual Behavior Profiling for Online Banking Fraud Detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 3, pp. 643 - 654, Aug. 2021.
[40] Y. Xie, G. Liu, C. Yan, C Jiang and M Zhou, "Time-aware attention-based gated network for credit card fraud detection by extracting transactional behaviors," *IEEE Trans. Computat. Social Syst.*, early access, doi: 10.1109/TCSS.2022.3158318.
[41] Q. Yang, C. Wang, C. Wang, H, Teng and C Jiang, "Fundamental Limits of Data Utility: A Case Study for Data-Driven Identity Authentication," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 2, pp. 398-409, Aug. 2021.
[42] J. Liang, Y. Tang, R. Hare, B. Wu and F. Wang, "A Learning-Embedded Attributed Petri Net to Optimize Student Learning in a Serious Game," *IEEE Trans. Computat. Social Syst.*, early access, doi: 10.1109/TCSS.2021.3132355.
[43] L. He, G. Liu and M. Zhou, "Petri-Net-Based Model Checking for Privacy-Critical Multiagent Systems," *IEEE Trans. Computat. Social Syst.*, early access, doi: 10.1109/TCSS.2022.3164052.
[44] F Zhao, D. Xiang, G Liu and C Jiang, "A New Method for Measuring the Behavioral Consistency Degree of WF-Net Systems," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 2, pp. 480--493, Sep. 2022.
[45] G.J. Liu, *Petri Nets: Theoretical Models and Analysis Methods for Concurrent Systems.* Singapore, Singapore, Springer, Nov. 2022, pp. 123-165.

WangYang Yu received the Ph.D. degree from Tongji University, Shanghai, China, in 2014. He is currently an Associate Professor with the School of Computer Science, Shaanxi Normal University, Xi'an, China. His research interests include the theory of Petri nets, formal methods in software engineering, and artificial intelligence.

YaDi Wang is a postgraduate student with the School of Computer Science, Shaanxi Normal University, Xi'an, China. Her research interests include the theory of Petri nets, process mining, online transaction systems, formal methods in software engineering, and artificial intelligence.

Lu Liu is the Head of School of Computing and Mathematical Sciences at the University of Leicester, UK. Professor Liu received his PhD degree from Surrey Space Centre at the University of Surrey, UK. His research interests are in the areas of data analytics, service computing, sustainable computing and the Internet of Things. He has over 250 scientific publications in reputable journals, academic books and international conferences. Professor Liu has secured many research projects which are supported by research councils, BIS, Innovate UK, British Council and leading industries. He received the Vice-Chancellor's Award for Excellence in Doctoral Supervision in 2018, BCL Faculty Research Award in 2012 and the Promising Researcher Award in 2011. He has been the recipient of 7 Best Paper Awards from international conferences and was invited to deliver 8 keynote speeches at international conferences. Professor Liu is a Fellow of BCS (British Computer Society). He is currently serving as an Associate Editor for Peer-to-Peer Networking and Application (PPNA) and Big Data Mining and Analytics (BDMA). He has chaired over 20 international conferences in the areas of Data Science AI Cloud Computing and the Internet of Things.

YiSheng An received the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2007, respectively. He is an IEEE Member, and a Professor with the Department of Computer Science and Engineering, School of Information Engineering, Chang'an University, Xi'an. His research interests include Internet of Vehicles, intelligent transportation systems and distributed information systems.

Bo Yuan received the BEng and PhD degree in computer science from the Tongji University, Shanghai, China in 2011 and 2017, respectively. He is currently a Lecturer in Computer Science with the School of Computing and Mathematical Sciences, University of Leicester, UK. His research interests include Distributed Networks, Artificial Intelligence, Internet of Things, Federated Learning, and Edge Computing.

John Panneerselvam is a Lecturer in Informatics at the University of Leicester, UK. John received his PhD in Computing from the University of Derby in 2018 and an MSc in advanced computer networks in 2013. He is an active member of IEEE and British Computer Society, and a HEA fellow. His research interests include cloud computing, fog computing, Internet of Things, big data analytics, bioinformatics, and P2P computing.

Scientific Digest
An International publication
Journal of Applied Engineering

ISSN: 2348-4802

A MULTI-VIEW FRAUD DETECTION APPROACH FOR MULTI-PARTICIPANT E-COMMERCE TRANSACTIONS

GUIDE : K.Vaddi Kasulu M.Tech, ph.D
Professor and Head Of Department CSE-AI & DS
Eluru College of Engineering and Technology

TEAM LEAD : P. CHENNARAO[1]
TEAM MEMBERS: K. RAMESH[2]
D. PRASANNA KUMAR[3]
SK. RIZWANA[4]
Eluru College of Engineering and Technology

1. INTRODUCTION

E-commerce has made shopping more convenient than ever, but with its rapid growth comes a serious challenge fraud. Online platforms are constantly battling fraudulent activities, from fake accounts to unauthorized transactions. Traditional fraud detection systems usually focus on just one aspect of a transaction, like payment details or user credentials. While this can catch some fraud, it often misses more complex schemes, especially in cases where multiple participants—buyers, sellers, and payment processors are involved.

To tackle this issue, our study introduces a Multi-View Fraud Detection Approach that looks at transactions from multiple angles. Instead of just checking basic details, our system analyzes user behavior, transaction history, device information, and network activity to spot suspicious patterns. Using machine learning and data fusion techniques, it improves fraud detection accuracy while reducing false alarms. Unlike conventional methods that might overlook subtle red flags, our approach uses anomaly detection, feature engineering, and ensemble learning to catch even the trickiest fraud attempts. We tested our method on real-world e-commerce data, and the results show it performs significantly better than traditional fraud detection systems. With fraud tactics constantly evolving, our approach helps online platforms stay one step ahead, ensuring safer and more reliable transactions for businesses and customers alike.

One of the biggest problems with older fraud detection systems is that they generate too many false positives, incorrectly flagging legitimate transactions as fraud. This can lead to frustrated customers and financial losses for businesses. As online transactions become more common, fraud detection needs to keep up with emerging threats. Our research aims to build smarter, faster, and more reliable fraud detection systems that help businesses stay ahead of fraudsters. By creating a more secure e- commerce environment, we can ensure safer and smoother transactions for both businesses and customers.

To achieve this, we employ Support Vector Machine (SVM), a powerful machine learning algorithm known for its ability to classify complex patterns efficiently. SVM helps distinguish between legitimate and fraudulent transactions by identifying subtle anomalies that may go unnoticed in traditional methods By implementing this multi-view approach, we aim to enhance security, minimize financial losses, and build a more trustworthy e-commerce ecosystem. With fraud becoming more sophisticated, businesses need smarter data-driven solutions and this project is a step in that directios

2. ABSTRACT

The rise of e-commerce has brought immense convenience to consumers and businesses, but it has also opened doors to fraudulent activities. Traditional fraud detection systems often rely on single-view analysis, where each transaction is examined in isolation. This approach, however, struggles to detect fraud that involves multiple participants, such as buyers, sellers, payment gateways, and logistics providers. As fraudsters develop more sophisticated tactics, there is a growing need for smarter, multi-perspective fraud detection methods.

This project introduces a Multi-View Fraud Detection Approach that analyzes transactions from different perspectives, considering user behavior, transaction history, and network interactions. Instead of treating each transaction as an independent event, our model examines relationships and patterns across multiple entities to improve fraud detection accuracy. We utilize Support Vector Machine (SVM), a powerful machine learning algorithm known for its efficiency in handling high- dimensional data and complex classification problems.

By leveraging multi-view learning, the proposed approach enhances fraud detection efficiency while reducing false positives. This results in a more secure and trustworthy e-commerce ecosystem, protecting businesses from financial losses and ensuring safer transactions for users. As fraudulent Stechniques evolve, adopting intelligent, data- driven solutions becomes essential. This project contributes to the advancement of fraud detection by offering a robust, adaptive, and scalable model that can help businesses combat fraud in real time.

3. EXISTING SYSTEM

Traditional fraud detection in e-commerce relies on rule-based systems and single-perspective machine learning models, which are limited in identifying complex fraud schemes. Rule-based systems use predefined conditions to flag fraud but struggle against evolving fraud tactics. Fraudsters can easily bypass these rules, leading to high false positives that block legitimate transactions.Machine learning models improve detection but often analyze only one aspect, like transaction history, ignoring multi-participant fraud. Coordinated fraud involving fake buyers, sellers, and payment processors remains undetected due to this narrow approach.

Many fraud detection systems operate in batch mode, identifying fraud after transactions are completed instead of preventing it in real-time. Limited data sources and lack of multi-perspective analysis reduce fraud detection accuracy. As a result, fraud remains a growing challenge in e-commerce transactions.

A multi-view fraud detection system is needed to analyze transactions from multiple perspectives, detect fraud in real time, and reduce false positives. This approach enhances security and ensures safer e-commerce transactions.

DISADVANTAGES OF EXISTING SYSTEM

- Security Single-Perspective Limitation – Existing fraud detection methods focus on only one aspect, such as user behavior or transaction history, making them ineffective against sophisticated fraud patterns.
- Lack of Real-Time Analysis – Most existing systems detect fraud after the transaction is completed, leading to delayed responses and potential financial losses.
- Scalability Issues – As e-commerce platforms grow, traditional fraud detection systems struggle to process large volumes of transaction data efficiently.

4. PROPOSED SYSTEM

Fraud in e-commerce transactions is becoming more sophisticated, making it harder for traditional fraud detection methods to keep up. Most existing systems rely on single-view analysis, where each transaction is examined in isolation. However, fraudsters often manipulate multiple aspects of a transaction, making it essential to analyze data from multiple perspectives.

Our project introduces a Multi-View Fraud Detection Approach that considers multiple factors—user behavior, transaction history, and network interactions—to enhance fraud detection accuracy.

The system works by first collecting and preprocessing data from different sources, ensuring that the information is clean and structured. It then applies multi-view learning to analyze transactions from different perspectives.

User behavior is monitored for suspicious activities, transaction histories are compared for irregularities, and network interactions are checked for hidden fraud patterns.

When a suspicious transaction is detected, the system triggers real-time alerts, allowing security teams to take immediate action. To further improve accuracy, techniques like cross-validation, hyperparameter tuning, and ensemble learning are applied.

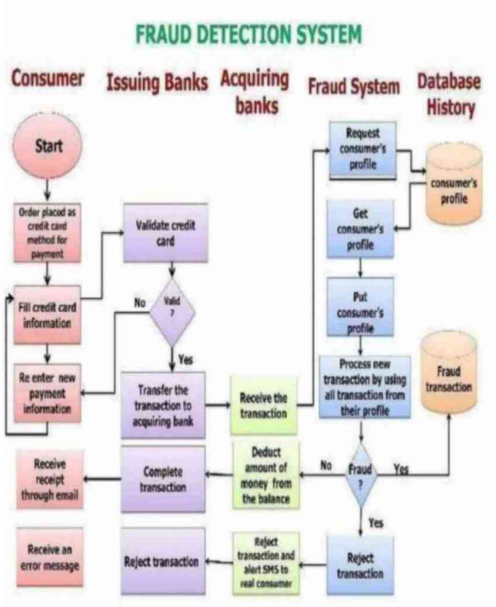
By using a multi-view learning approach, this system enhances fraud detection accuracy, reduces false positives, and ensures a safer online shopping experience. It is scalable, adaptable, and capable of evolving with emerging fraud techniques, making it a robust solution for modern e-commerce security.

Moreover, this system is designed to be scalable and adaptable, making it suitable for both small and large e-commerce platforms. As fraud techniques continue to evolve, the model can be continuously trained with new data, improving its ability to detect emerging threats

ADVANTAGES PROPOSED SYSTEM

1. Higher Fraud Detection Accuracy – By using a multi-view learning approach, the system analyzes transactions from multiple perspectives, significantly improving fraud detection accuracy.
2. Reduced False Positives – Unlike traditional methods that often block legitimate transactions, this approach ensures genuine users are not wrongly flagged as fraudsters.
3. Real-Time Detection & Alerts – The system processes transactions in real-time, allowing for immediate action against fraudulent activities, reducing financial losses.
4. Scalability & Adaptability – The model can handle large-scale e-commerce transactions and adapt to evolving fraud patterns, making it suitable for businesses of all sizes.
5. Enhanced Security & Trust – By effectively identifying and blocking fraudulent transactions, the system helps build customer confidence and trust in e-commerce platforms.
6. Machine Learning-Based Intelligence – The use of Support Vector Machine (SVM) ensures precise classification of fraud while allowing the model to continuously learn and improve.
7. Comprehensive Multi-Participant Analysis – Unlike single-view methods, this system examines buyers, sellers, payment gateways, and logistics providers to uncover hidden fraud networks.
8. Automation & Efficiency – Reduces the need for manual fraud detection efforts, saving time and operational costs for e-commerce businesses.

5. SYSTEM ARCHITECTURE



In this fraud detection system, the consumer steps forward as an enthusiastic explorer, inputting credit card details to initiate a purchase. The issuing bank stands as a vigilant guardian, swiftly verifying the card's authenticity. Once approved, the acquiring bank takes on the role of a charismatic conductor, orchestrating the payment flow with finesse. Simultaneously, the fraud detection system, a watchful detective, monitors every transaction for suspicious patterns. Drawing on records from the database, the memory keeper, it compares current actions with historical data.

Should any anomalies arise, the detective raises an immediate alert and halts the transaction. Meanwhile, legitimate transactions are cleared to sail through, granting the consumer a seamless experience. The system then notifies the user about approvals or potential fraud, maintaining transparency. Working as a well- coordinated team, these characters ensure security without sacrificing convenience. Ultimately, this humanoid-like architecture provides a dynamic and reliable shield against fraudulent activities.

REFERENCES

1. A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90-113, Apr. 2016.
2. E. A. Minastireanu and G. Mesnita, "An analysis of the most used machine learning algorithms for online fraud detection," *Info. Econ.*, vol. 23, no. 1, 2019.
3. X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," *arXiv preprint*, vol. 1904, no. 10604, 2019, doi: 10.48550/arXiv.1904.10604.
4. L. Zheng et al., "Transaction fraud detection based on total order relation and behavior diversity," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 3, pp. 796-806, 2018.
5. Z. Li, G. Liu, and C. Jiang, "Deep representation learning with full center loss for credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 569-579, 2020.
6. I. M. Mary and M. Priyadharsini, "Online transaction fraud detection system," in *2021 Int. Conf. Adv. C. Inno. Tech. Engr. (ICACITE)*, 2021, pp. 14-16.
7. D. Choi and K. Lee, "Machine learning based approach to financial fraud detection process in mobile payment system," *IT Conv. P. (INPRA)*, vol. 5, no. 4, pp. 12-24, 2017.
   R. Sarno et al., "Hybrid association rule learning and process mining for fraud detection," *IAENG Int. J. C. Sci.*, vol. 42, no. 2, 2015.
8. Jans et al., "A business process mining application for internal transaction fraud mitigation," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13351-13359, 2011.
9. K. D. Febriyanti, R. Sarno, and Y. Effendi, "Fraud detection on event logs using fuzzy association rule learning," in *2017 11th Int. Conf. Info. Comm. Tech. Sys.*, Surabaya, Indonesia, 2017, pp. 149-154.
10. T. Chiu, Y. Wang, and M. Vasarhelyi, "A framework of applying process mining for fraud scheme detection," *SSRN Electronic Journal*, 2017, doi: 10.2139/ssrn.2995286.
11. W. Rui, S. Chen, X. Wang, and S. Qadeer, "How to shop for free online—Security analysis of cashier-as- a-service based web stores," in *Proc. SSP*, 2017.
12. R. A. Kuscu, Y. Cicekcisoy, and U. Bozoklu, *Electronic Payment Systems in Electronic Commerce*. Turkey: IGI Global, 2020, pp. 114–139.
13. M. Abdelrhim and A. Elsayed, "The effect of COVID- 19 spread on the e-commerce market: The case of the 5 largest e-commerce companies in the world," *SSRN 3621166*, 2020, doi: 10.2139/ssrn.3621166.
14. P. Rao et al., "The e-commerce supply chain and environmental sustainability: An empirical investigation on the online retail sector," *Cogent Bus. Manag.*, vol. 8, no. 1, pp. 1938377, 2021.
15. J. J. Stoop, "Process mining and fraud detection—A case study on the theoretical and practical value of using process mining for the detection of fraudulent behavior in the procurement process," M.S. thesis, ENS: University of Twente, Netherlands, 2012.
16. M. D. Leoni, W. M. Van Der Aalst, and B. F. V. Dongen, "Data-and resource-aware conformance checking of business processes," in *Int. Conf. Bus. Info. Sys.*, Springer, Berlin, Heidelberg, 2012, pp. 48-59.
17. K. Bömer and S. Rinderle-Ma, "Anomaly detection in business process runtime behavior—challenges and limitations," *arXiv preprint*, arXiv:1705.06659, 2017, doi: 10.48550/arXiv.1705.06659.
18. T. Chomyat and W. Premchaiswadi, "Process mining on medical treatment history using conformance checking," in *2016 14th Int. Conf. ICT Knowl. Eng. (ICT&KE)*, 2016, pp. 77-83.
19. K. D. Febriyanti, R. Sarno, and Y. Effendi, "Fraud detection on event logs using fuzzy association rule learning," in *2017 11th Int. Conf. Info. Comm. Tech. Sys.*, Surabaya, Indonesia, 2017, pp. 149-154.
20. S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C.

Author's Profile

GUIDE : K.VADDI KASULU M.tech(CSE)

Ph.d(CSE), Working as Associate Professor and Head Of the Department in Department of CSE – (AI&DS), Eluru College of Engineering and Technology, Eluru.

EMAIL : vaddi1229@gmail.com

P.CHENNARAO B.Tech with Specialization of Computer Science and Engineering(Artificial Intelligence and Data Science) in Eluru college of Engineering, Eluru.

EMAIL ; chennaraoparsa1@gmail.com

K.RAMESH B.tech with Specialization of Computer Science and Engineering (Artificial Intelligence and Data Science),in Eluru College of Engineering and Technology, Eluru.

EMAIL : ramesh1504k@gmail.com

D.PRASANNA KUMAR B.tech with Specialization of Computer Science and Engineering (Artificial Intelligence and Data Science) in Eluru College of Engineering and Technology, Eluru.

EMAIL: prasannakumardevarapalli1@gmail.com

SK.RIZWANA B.tech with Specialization of Computer Science and Engineering (Artificial Intelligence and Data Science) in Eluru College of Engineering and Technology, Eluru.

EMAIL : shaikrizwanasardar11@gmail.com