

GENERATIVE AI IN REAL-TIME E-COMMERCE FRAUDDETECTION: A COMPARATIVE AND ETHICAL ANALYSIS

¹PILLI NAGA SAI GANESH, ²Dr.R. RAMBABU REDDY

¹Student, ²Associate Professor

¹DEPARTMENT OF COMPUTER APPLICATIONS

¹²RAJAHMAHENDRI INSTITUTE OF ENGINEERING AND TECHNOLOGY, Rajahmahendry

Abstract—This paper will discuss generative AI models, such as GANs, VAEs, and a hybrid GAN-VAE, towards real-time fraud detection in e-commerce. GANs perform outstandingly in synthesizing fraudulent data, while VAEs detect complex patterns of fraud. The proposed hybrid method combines the respective strengths and achieves higher accuracy and adaptability. Also, issues regarding ethical concerns related to privacy in data and biases are taken care of, illustrating the future possibility of generative AI in upgrading the frameworks for fraud detection in ecommerce. **Keywords**—Generative AI, Generative Adversarial Networks, Variational Autoencoders, Machine Learning, E-Commerce Fraud

Index Terms—Generative Ai, Real-Time, E-Commerce ,Fraud detection, Ethical Analysis

I. Introduction

This paper will discuss generative AI models, such as GANs, VAEs, and a hybrid GAN-VAE, towards real-time fraud detection in e-commerce. GANs perform outstandingly in synthesizing fraudulent data, while VAEs detect complex patterns of fraud. The proposed hybrid method combines the respective strengths and achieves higher accuracy and adaptability. Also, issues regarding ethical concerns related to privacy in data and biases are taken care of, illustrating the future possibility of generative AI in upgrading the frameworks for fraud detection in ecommerce. **Keywords**— Generative AI, Generative Adversarial Networks, Variational Autoencoders, Machine Learning, E-Commerce Fraud information to improve cybersecurity and prevent attacks. Digital forensics investigations also help the cybersecurity community improve security and identify new threats. Digital evidence preservation is a digital forensics premise[9]. Evidence can be manipulated, removed, or destroyed in a digital environment, thus strict protocols are needed to preserve it. Digital forensic professionals utilize specialized tools and techniques to generate forensic copies of digital media, preserving the original data for court use[10]. Create bit-for-bit replicas of storage devices like hard drives and mobile phones and use cryptographic hashes to validate their authenticity. The analysis step of digital forensics is probably the most complex and important. It entails searching digital evidence for caserelated information. Digital forensic investigators employ many software tools to recover deleted files, analyze system logs, network traffic, and decrypt encrypted data. They use forensic accounting to track money transactions and criminal patterns[11]. Digital forensics professionals meticulously reconstruct the events leading up to and after the malicious activity, offering a complete

picture. Digital data volume is a major difficulty in digital forensics. Today's linked world generates and stores vast volumes of data constantly[12]. Individuals and businesses leave huge digital traces, including emails, texts, and social media. Digital forensic algorithms to find pertinent data in this massive data set. Digital forensics' chain of custody assures digital evidence's integrity and admissibility in court[13]. To maintain a clear path from evidence collection to court presentation, proper recording and handling are crucial. This assures that the evidence has not been altered during the inquiry, preserving its credibility and legality[14]. Digital forensics is multidisciplinary and requires several skills. In this industry, professionals have expertise in computer technology, law enforcement, cybersecurity, and forensic accounting. Specialized training equips them to perform effective digital investigations[15].

II. SYSTEM ANALYSIS

The existing e-commerce fraud detection systems primarily rely on rule-based engines or traditional machine learning models like Logistic Regression, Decision Trees, or Support Vector Machines. These systems flag fraud based on predefined thresholds (e.g., unusually high amounts, blacklisted IPs) or static training data. While effective in detecting known fraud patterns, they often fail when confronted with new or evolving tactics. Moreover, they do not generate synthetic data or adapt to changing fraud patterns without frequent manual updates. These systems lack contextual understanding and struggle to detect complex or subtle anomalies, especially in real-time scenarios.

Disadvantages:

1. Rigid and rule-dependent, prone to high false positives Poor adaptability to evolving fraud techniques
2. Requires manual feature engineering and model retraining Cannot generate synthetic fraud data for imbalanced datasets Fails to handle real-time latency demands effectively
3. Lacks insight into ethical concerns like demographic bias or user fairness

PROPOSED SYSTEM

The proposed system integrates Generative AI models (GAN, VAE, and Hybrid GAN-VAE) within a Django-based web platform for real-time e-commerce fraud detection. The system improves fraud detection accuracy by using GANs to synthesize fraud data, VAEs to capture complex latent patterns, and a hybrid model to combine both strengths. It supports dynamic user interaction through a user-friendly interface, dataset analysis, and real-time fraud prediction. Ethical considerations such as data privacy and regional bias analysis are also incorporated. The system further integrates Google Gemini to generate analytical insights from uploaded data, assisting in fraud pattern interpretation and decision-making.

Advantages:

1. Uses GANs to generate synthetic fraudulent transactions for balanced datasets
2. VAE and Hybrid models enhance anomaly detection accuracy and reduce false positives
3. Real-time fraud prediction with low-latency support
4. Automated dataset analysis with AI-generated insight (via Gemini API)
5. Built-in visualization (correlation heatmaps, model comparison charts)

FUNCTIONAL REQUIREMENTS

A Data Preprocessing Module is necessary to clean the tweets by removing noise such as links, special characters, stop words, and emojis, followed by standard Natural Language Processing (NLP) steps such as tokenization and stemming. Once preprocessed, the tweets undergo Sentiment Analysis to classify them as positive, negative, or neutral, which helps in identifying emotional tone. Further, the tweets are

analyzed for Depression Classification using machine learning algorithms like Naïve Bayes and NBTree, which categorize them as depressive or non-depressive.

The system should also support Model Comparison, allowing users to compare the performance of different algorithms based on accuracy, precision, recall, and F1-score. For better interpretation of results, Data Visualization tools such as word clouds, sentiment graphs, bar charts, and pie charts will be generated. Optionally, users can input a custom tweet into the system to predict its depressive status, adding a layer of personalization. Additionally, the system should enable Report Generation of the analysis for documentation or further review. Finally, Security and Privacy are critical; hence, user data will be anonymized and handled ethically to ensure compliance with privacy standards.

Graphical User Interface (GUI):

A user-friendly dashboard/interface to allow users, researchers, and healthcare professionals to interact with the system and view insights.

1. Twitter Data Extraction:

The system must fetch or import tweet data using Twitter API or pre-collected datasets containing user tweets relevant to mental health.

2. Data Preprocessing Module:

The system should clean and preprocess tweet data by removing stop words, links, special characters, emojis, and applying normalization techniques like tokenization and stemming.

3. Sentiment Analysis:

It should analyze the sentiment of each tweet and categorize it into positive, negative, or neutral.

4. Depression Classification:

The system must classify tweets as depressive or non-depressive using implemented machine learning algorithms such as Naïve Bayes and NBTree.

5. Model Comparison:

The system should calculate and display the performance metrics (accuracy, precision, recall, F1score) for both algorithms for analysis and comparison.

6. Data Visualization:

It should generate and display visual outputs like pie charts, sentiment graphs, tweet word clouds, and bar charts for sentiment distribution.

7. User Input Option :

Allow user to enter a custom tweet and predict if the text shows depressive traits.

8. Security & Privacy:

Ensure anonymization of user data to comply with privacy and ethical standards during processing and display.

PERFORMANCE REQUIREMENTS

The system is designed to efficiently process and analyze large volumes of real-time Twitter data for detecting signs of depression. The following performance requirements are essential to ensure optimal functionality and user experience:

Real-Time-Tweet-Processing

The system should be able to fetch and preprocess tweets in near real-time using the Twitter API without significant latency. The average tweet retrieval and preprocessing time should not exceed 2–3 seconds per batch.

High-Accuracy-and-Precision-in-Classification

The machine learning models (Naïve Bayes and NBTree) should achieve an accuracy of at least 75% in classifying depressive and non-depressive tweets. The model should also maintain balanced precision and recall to avoid false positives or false negatives.

Fast-Model-Inference-Time

Once trained, the models should predict the sentiment of a new tweet in less than 1 second, ensuring quick response for large datasets.

Scalability

The system should be scalable to handle increasing tweet volumes over time. It should be able to process at least 10,000 tweets per hour without system crashes or memory issues.

User-Friendly-Dashboard

The dashboard should load within 2–3 seconds, and visualizations like sentiment distribution graphs, pie charts, and word clouds should render smoothly without lag

III. System Design

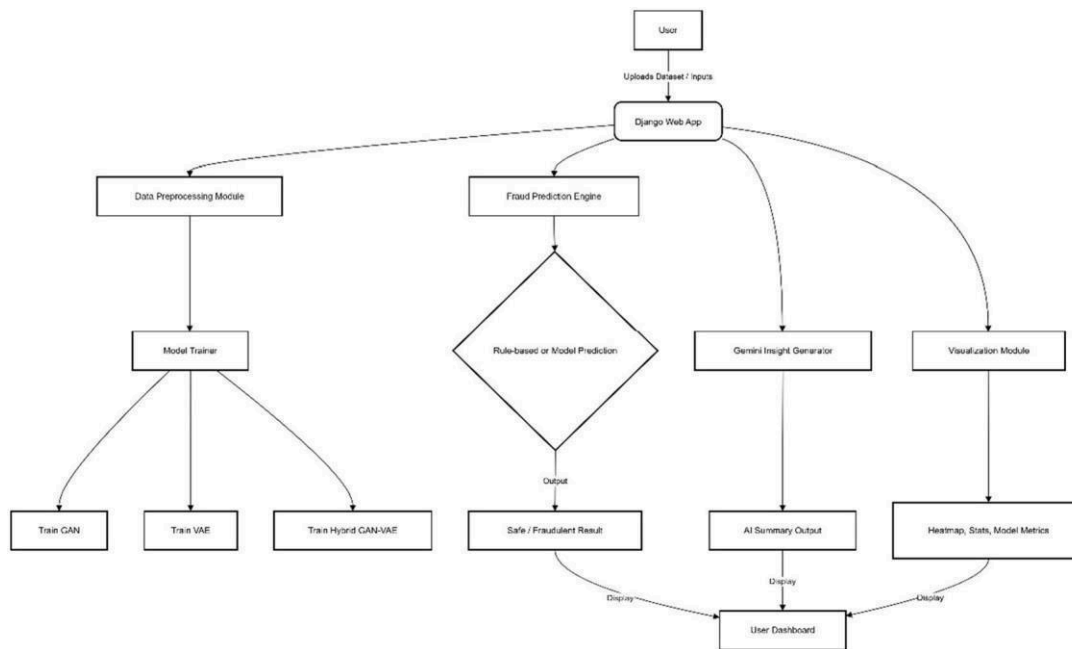
System design is a crucial phase in software development where the overall structure and functional flow of the system is defined. It translates the system requirements into a blueprint for building the system. In this project, the system design involves the creation of data models, process models, and interaction diagrams to ensure the application meets both functional and non-functional requirements. The primary objective of the system design phase is to provide a clear understanding of how the components of the depression detection system will work together. This includes how data will be collected from Twitter, how it will be preprocessed and analyzed, and how results will be presented to the end-user.

The design phase consists of:

- **System Architecture:** Defines the structure of the entire system including the client-server model, APIs, and databases.
- **E-R Diagram:** Illustrates the data entities and relationships.
- **Data Flow Diagram (DFD):** Describes how data moves through the system.
- **UML Diagrams:** Visualizes the system's structure and behavior using standard modeling techniques such as Use Case, Class, Sequence, and Activity diagrams.
- **Data Dictionary:** Provides definitions of data elements used throughout the system.

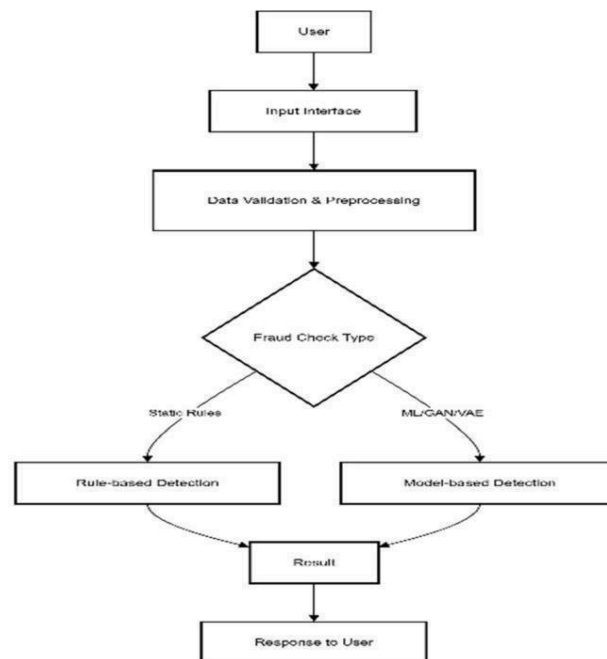
By using these design models, the development process becomes more efficient, understandable, and scalable, ensuring that the system is robust, maintainable, and aligned with user expectations

IV. SYSTEM ARCHITECTURE



4.1 DATA FLOW DIAGRAMS

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other nonsoftware systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

V. CONCLUSIONS

This study aims to diversify the e-commerce ecosystem while enhancing resilience to casual attacks through the use of generative AI models, such as GANs, VAEs, and hybrid GAN-VAE architectural integration. All things considered, each model made a contribution to the identification of recognized fraud. In other words, though generative AI is a great milestone in the war against e-commerce fraud, the search for improvement and combination of models remains a constant to effectively utilize these capabilities in the real time scenarios and on extensive sizes. Help develop the right voice, including real time monitoring systems to improve prevention of future styled fraud that has not been experienced, and advocate for moral perspectives that build confidence and acceptance. As a result, AI generative has the ability to revolutionize the standards of online fraud detection, where e-commerce operators will provide more adaptive, comprehensive, and protective mechanisms against fraud in case incidents arise

References

- [1] A. Jain and S. Raghavan, "Fairness in machine learning: A survey," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–35, 2021. S. Huang and W. Chen, "Ethical considerations in AI and machine learning: The role of data governance," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1647–1670, 2020.
- [2] survey," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–35, 2021. S. Huang and W. Chen, "Ethical considerations in AI and machine learning: The role of data governance," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1647–1670, 2020.
- [3] H. Q. Nguyen and T. T. Nguyen, "Detecting fraud in e-commerce: A review of recent advances and future directions," *Expert Systems with Applications*, vol. 177, p. 114914, 2021.
- [4] S. Tyagi, D. Kumar, and S. Kumar, "Understanding the nitty gritty of software reliability and its testing procedures: A different approach," *Journal of Information and Optimization Sciences*, vol. 38, no. 6, pp. 971–988, 2017.
- [5] Y. Li and J. Wang, "Generative models for e-commerce fraud detection: A comparative study," *Journal of Computer Science and Technology*, vol. 36, no. 3, pp. 505–520, 2021.
- [6] J. Zhou and Q. Chen, "E-commerce fraud detection using machine learning: A review," *Journal of Systems and Software*, vol. 151, pp. 134–150, 2019.
- [7] "Fairness in machine learning: Lessons from political philosophy," in *Proc. 2018 Conf. Fairness, Accountability, and Transparency (FAT)*, 2018.
- [8] J. Choi and K. J. Kim, "The use of generative adversarial networks in e-commerce fraud detection: A systematic review," *Journal of Business Research*, vol. 139, pp. 479–489, 2022.
- [9] H. Sarker and D. O'Sullivan, "Exploring the use of generative models for fraud detection: Current trends and future directions," *Journal of Risk and Financial Management*, vol. 16, no. 3, p. 101, 2023.
- [10] S. Tyagi, D. Kumar, and S. Kumar, "Reliability based solution to the decision making dilemma in a software environment," *Journal of Statistics and Management Systems*, vol. 22, pp. 627–634, 2019, doi: 10.1080/09720510.2019.1611226.