Preventing and Mitigating Unauthorized Access in Banking Using Artificial Intelligence

Group 9: Course Project Report

CSE 543: Information Assurance and Security [Spring 2025]

Group Members:

- **Nimesh Bali Yadav** **(Group Leader)**
- **Dan Lander** **(Deputy Leader)**
- **Bright Manu** **(Member)**
- **Mann Anil Vora** **(Member)**
- **Arvind Mahendran** **(Member)**
- **Thomas Tung** **(Member)**

Table of Contents:

1. Introduction

1.1 Motivation and Background

The increasing digitalization of banking operations has elevated the risk of cyber threats, particularly those involving unauthorized access to sensitive financial accounts. Among the most pervasive and financially damaging forms of such threats is credit card fraud, which continues to evolve in complexity and scale. As banking services become more interconnected through mobile, online, and core banking systems, the attack surface for malicious entities expands correspondingly.

Conventional rule-based fraud detection systems, though still in use, lack the adaptability and scalability required to counter modern fraud patterns. These systems operate on predefined static rules, which are often inadequate when facing novel fraud techniques that mimic legitimate customer behavior. Furthermore, the rigid structure of rule-based models leads to high false positive rates and limited capacity for learning from new transaction patterns.

To address these limitations, the use of Artificial Intelligence (AI) and Machine Learning (ML) techniques has gained momentum in the field of fraud detection. These data-driven approaches can identify complex, nonlinear relationships and behavioral anomalies in transactional data. Unlike traditional models, AI-based systems are capable of continuous learning and adaptation, which makes them well-suited for combating emerging fraud strategies in real-time environments.

This project is motivated by the need to enhance the effectiveness of fraud detection systems in mitigating unauthorized access to financial systems. By leveraging advanced ML algorithms such as Support Vector Machines (SVM), Principal Component Analysis (PCA), and Hidden Markov Models (HMM), the project aims to create a more robust detection framework. These techniques are particularly relevant due to their respective strengths—SVMs in classification, PCA in dimensionality reduction, and HMMs in modeling sequential transaction behavior.

Recent advancements highlight the value of modern AI and behavioral analysis in enhancing fraud detection accuracy and system transparency. This project incorporates these elements to align with both technical requirements and regulatory expectations.

1.2 Goals and Scope of Study

The primary objective of this project is to design a credit card fraud detection system that leverages artificial intelligence to improve the identification of unauthorized and suspicious transactions. The solution is intended to be applicable to a wide range of digital banking platforms, including core, online, and mobile banking systems.

The project is structured around three key focus areas:

- **Model Selection and Implementation:** The study will utilize supervised and unsupervised learning methods, including Logistic Regression, k-Nearest Neighbors (k-NN), Autoencoders, and the Forward Algorithm for Hidden Markov Models (HMMs). These models will be tested for their ability to distinguish fraudulent from legitimate transactions, especially under real-world class imbalance conditions.

- **Application Relevance:** The designed fraud detection system will be evaluated in the context of modern banking infrastructure. Special attention will be given to scalability and performance in high-volume transaction environments, with a goal of producing actionable alerts in real time.

- **Research Alignment:** The project will examine current and emerging research in credit card fraud detection, particularly in the areas of behavioral analytics and modern AI. The intention is to integrate these concepts into the system to enhance both predictive accuracy and trustworthiness.

This project is expected to provide insights into AI-based fraud detection techniques, key fraud indicators, and documented findings for future improvements.

2. Summary of Accomplishments of the Project

- Evaluated a range of supervised learning models—Random Forest, Decision Tree, Support Vector Machine (SVM), and Neural Network—for their applicability to credit card fraud detection in real-world financial systems and performance were analyzed.
- Explored ensemble learning approaches such as bagging and boosting, including algorithms like XGBoost, LightGBM, CatBoost, and AdaBoost. These models demonstrated strong generalization under class-imbalanced conditions and consistently outperformed baseline classifiers by reducing both false positives and false negatives.
- Analyzed behavioral profiling strategies to detect anomalies in temporal transaction patterns. Using fixed sliding windows and distance-based metrics, such as Euclidean and permutation-aligned distances, the system identified subtle behavioral shifts indicative of unauthorized access.
- Examined the class imbalance problem prevalent in fraud detection datasets, where fraudulent events form a small minority. Approaches like random under-sampling and cost-sensitive learning were reviewed alongside performance metrics such as Area Under Curve - Precision Recall (AUC-PR) and F1-score to assess their impact on minority-class classification.
- Studied graph-based fraud detection methods by transforming transactional data into graph structures, enabling the use of Graph Neural Networks (GNNs), Graph Convolutional Networks (GCNs), and Graph Attention Networks (GATs) to model complex relationships between users, merchants, and transactions.
- Implemented a custom GAT model with multi-feature aggregation and node-level attention to enhance fraud classification in graph-structured data. The model achieved high performance in detecting fraudulent nodes based on contextual relational information and transactional behavior.
- Developed a standalone Hidden Markov Model (HMM) for identifying deviations in sequential transaction data. By computing transition probabilities and analyzing observed

output transactions, the model effectively flagged transaction chains that diverged from learned behavioral spending patterns.

- Extended the HMM framework to support real-time spending category validity updates and high-value transaction threshold monitoring. These enhancements improved fraud detection accuracy in dynamic environments with evolving user behavior.
- Explored deep learning models such as Recurrent Neural Networks (RNNs), Transformers, and Bidirectional Encoder Representations from Transformers (BERT) for sequence modeling of financial transactions. These architectures captured temporal dependencies and behavioral trends over long transaction histories.
- Examined the role of Horizontal Federated Learning (HFL) in preserving data privacy while enabling collaborative model training across financial institutions. HFL supports decentralized learning without raw data exchange, making it suitable for regulatory-sensitive environments.
- Analyzed hybrid models that combine supervised learning with unsupervised anomaly detection and time-series analysis. These models were found to be robust against concept drift and capable of identifying unknown fraud patterns not seen during training.
- Contributed to the development and evaluation of seven fraud detection models, ranging from traditional algorithms to advanced ensemble methods. Random Forest, CatBoost, and XGBoost consistently delivered top-tier performance in terms of AUC, recall, and F1-score on highly imbalanced datasets.
- Proposed the integration of real-time processing tools (e.g., Apache Kafka, Flink) and online learning techniques to continuously adapt the model to new fraud patterns. Long Short-Term Memory (LSTM) and Transformer models were highlighted for their suitability in sequential fraud prediction.
- Reviewed layered fraud detection architectures that integrate rule-based classifiers, temporal anomaly profiling, and graph-based reasoning. These hybrid frameworks, as studied in relevant research papers, offer comprehensive detection across transactional, behavioral, and relational fraud surfaces and were recommended for real-world deployment.

3. Responsibilities and Accomplishments of each Group Member

- The successful completion of this project, Preventing and Mitigating Unauthorized Access in Banking using Artificial Intelligence, was the result of consistent collaboration, appropriate task assignment, and a shared academic commitment to meeting the goals of the project. Throughout the duration of the course project, each member of the team contributed meaningfully to both the technical development and the final completion of the project.
- Responsibilities were allocated based on individual areas of interest and expertise, while also allowing for flexibility as the project evolved. Regular meetings were held to discuss progress, reassign tasks as needed, and maintain alignment with our overall objectives. These sessions facilitated open communication, collective problem-solving, and ensured that all members remained actively involved at each stage of the project.
- All team members participated in reviewing their assigned in-depth papers, discussing emerging ideas, and contributing to the implementation and evaluation of various fraud detection approaches. Contributions included exploring various machine learning techniques, assisting with model design, and writing technical sections of the report.
- This project served as an opportunity not only to apply concepts learned in class but also to deepen our understanding of real-world applications of artificial intelligence in financial security. The experience strengthened our ability to work collaboratively on complex topics, think critically, and contribute meaningfully to a real life application such as detecting credit card fraud.
- The subsections that follow provide a summary of each group member's individual responsibilities and accomplishments for this project.

3.1 Nimesh Bali Yadav (Group Leader)

- Studied reference papers: [1], [2], [3], [4], [5], [6], [7] and [43] related to the computer security problem of credit card fraud and machine learning techniques for fraud detection

- Completed weekly reports and helped evaluate team members' individual weekly and in-depth reports.
- Conducted thorough research and found relevant papers [1], [2], and [3] that explore various methods for credit card fraud detection using machine learning and AI-based systems.
- Completed 3 in-depth reports for papers [1], [2], and [3] listed in the references section of this final report, analyzing their methodologies, findings, and relevance to improving fraud detection techniques.
- Assisted in coordinating weekly tasks and clarifying instructions on Slack to ensure timely progress of the project.
- Contributed to simulating and evaluating various machine learning models for fraud detection, with a specific focus on applying and improving algorithms discussed in the referenced papers.
- Analyzed and compared different fraud detection methods, including the effectiveness of supervised and unsupervised learning algorithms as mentioned in papers [1], [2], and [3].
- Provided feedback on the project report, specifically sections discussing machine learning-based fraud detection algorithms and potential future work on enhancing the model with advanced techniques like blockchain or hybrid learning methods**.**

3.2 Dan Lander (Deputy Leader)

- Studied reference papers [8], [9], [10], [11], [12], [13] and [14] related to improving Credit Card Fraud Detection using Hidden Markov Models and/or Blockchain Technology.
- Completed weekly reports, helped evaluate other team member’s individual weekly reports, peer evaluated my assigned in-depth reports, helped to coordinate weekly tasks, and clarify instructions on Slack.
- Completed 3 in-depth reports for the papers [8], [9] and [12] listed in the references section of this final report.
- Completed coding and testing a simplified standalone Hidden Markov Model (HMM)

Detection Algorithm in python using the framework of the HMM Detection Algorithm presented in papers [8] and [9].

- Completed simulating and tabulating output results for my coded standalone HMM Detection Algorithm using various chain lengths and drop threshold values including a predetermined context of valid truth labels for each spending category.
- Completing analyzing the output results of my coded standalone HMM Detection Algorithm in order to determine the best performing and most accurate drop threshold/chain length combinations.
- Contributed to the final project report by completing my individual methods/ results sections for the HMM Detection Algorithm presented in papers [8] and [9] including a potential future work section for incorporating potential Blockchain Technology as presented in paper [12].

3.3 Bright Manu

- Studied reference papers [15], [16], [17], [18], [19], [20], [21] and [44] related to graph-based learning techniques for credit card fraud detection.
- Completed all weekly reports and assigned tasks, evaluated team members' individual weekly and in-depth reports.
- Completed three (3) in-depth study reports on [19], [20] and [21].
- Completed coding from scratch the Feature Aggregation GAT based model as presented in the paper [21].
- Performed simulation experiments to assess the model's performance in detecting fraudulent transactions.
- Combined, presented and analyzed the results from the simulation experiments in plots and tables.
- Contributed to the final project report**.**

3.4 Mann Anil Vora

- Studied reference papers [22], [23], [24], [25], [26], [27] and [28] related to ensemble machine learning models for credit card fraud detection.
- Participated actively in weekly team meetings and discussions, ensuring alignment with project milestones and addressing task-level blockers in coordination with the team lead.
- Completed weekly reports, helped evaluate other team member's individual weekly reports, peer evaluated my assigned in-depth reports, helped to coordinate weekly tasks, and clarify instructions on Slack.
- Assisted in reviewing and providing constructive feedback on peer reports, ensuring clarity, relevance, and academic rigor across submissions.
- Conducted supplementary research to strengthen the project's foundation by identifying and annotating research papers relevant to ensemble learning, real-time fraud detection, and synthetic data generation techniques used in our implementation [25].
- Contributed to the development and validation of seven machine learning models including Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost, CatBoost, and LightGBM by performing data preprocessing, hyperparameter tuning, and evaluation metric analysis [27] [28].
- Collaborated in the preparation of precision-recall visualizations, SHAP interpretation plots, and confusion matrix summaries to support the report's analysis sections.
- Drafted portions of the in-depth project report—particularly the modeling pipeline, evaluation, and methodology analysis—aligned with ensemble learning strategies discussed in scholarly research.
- Proposed and documented future enhancements based on recent academic developments, including the integration of LSTM networks and hybrid anomaly detection techniques for long-term scalability and adaptability of the fraud detection system [24].
- Completed in-depth reports on [26], [27] and [28] .
- Completed high-level reports on [22], [23], [24] and [25].

3.5 Arvind Mahendran

- Studied reference papers [29], [30], [31], [32], [33], [34] and [35] related to analyzing different machine learning methods for credit card fraud detection.
- Authored Sections 1, 2, 3, 4.1, 4.6 and 5 of the Final Group Project Report in their entirety.
- Weekly reports were completed, and peers conducted evaluations of the detailed submissions.
- Completed in-depth reports on [29], [30], [31] and [32].
- Completed high-level reports on [33], [34] and [35].
- Incorporated key insights from a comparative analysis of Machine Learning and Deep Learning approaches utilized for detecting unauthorized access in banking systems.

3.6 Thomas Tung

- Studied reference papers [36], [37], [38], [39], [40], [41] and [42] related to the usage of deep learning for credit card fraud detection.
- Completed Weekly reports, peer evaluated in-depth reports.
- Completed in-depth reports for [37], [41], [42].
- Authored Sections 4.7 of the final group project report examining different deep learning models and using federated learning for these models to have a bigger training set.

4. Detailed Results of Group Project

4.1 Overview of Banking in Artificial Intelligence

As financial systems continue to digitize, artificial intelligence (AI) has emerged as a transformative technology across all facets of banking. From streamlining customer engagement through intelligent virtual assistants to optimizing loan approvals with predictive analytics, AI is driving unprecedented gains in efficiency, personalization, and scalability. Among these applications, fraud detection—particularly concerning unauthorized access—has become a strategically critical area of research and implementation. This section introduces the role of AI in modern banking and outlines the prevailing security challenges that necessitate the development of more robust and intelligent frameworks for detecting credit card fraud.

4.1.1 What is Banking in Artificial Intelligence?

Banking in the context of artificial intelligence refers to the integration of AI-driven technologies into the core operations of financial institutions. These applications span a wide spectrum, including automated customer support, risk assessment, portfolio management, credit scoring, and fraud detection. Unlike traditional rule-based systems, AI models are capable of learning from historical and real-time data to make more informed and accurate real time decisions about whether a particular transaction may be fraudulent or not.

The adoption of machine learning (ML), deep learning, and natural language processing (NLP) in banking enables systems to detect trends, forecast user behavior, and identify anomalies at a scale and speed beyond the capabilities of manual analysis. AI is particularly well-suited for environments characterized by high data volume and velocity—such as real-time digital payment systems—where rapid and accurate decision-making is essential.

Fraud prevention—particularly in the context of cybersecurity—has emerged as one of the most critical applications of AI in the banking sector. As attackers develop increasingly sophisticated evasion techniques, static detection systems have proven insufficient. AI-powered fraud detection models are now designed to evolve continuously, detecting subtle behavioral anomalies and unauthorized activities that would otherwise have gone unnoticed by more traditional security mechanisms.

4.1.2 Security Challenges in Online Banking Today

Despite substantial advancements in digital banking infrastructure, the financial sector remains a primary target for cyber threats. Online banking systems are susceptible to various attack vectors, including phishing, credential stuffing, session hijacking, and vulnerabilities in APIs and third-party services. These attacks often lead to unauthorized access to financial systems or customer accounts, resulting in monetary theft, data breaches, and reputational harm.

Another central challenge in modern fraud detection lies in the ability of threat actors to closely mimic legitimate user behavior. Fraudsters increasingly deploy automation, adaptive attack strategies, and zero-day exploits to evade detection. Such behavior renders traditional rule-based models inadequate, as they are unable to keep pace with evolving attack techniques or detect context-sensitive anomalies.

To help combat these challenges, AI-based systems enable real-time monitoring of high-volume transactional data, allowing for the timely detection of emerging fraud patterns and the prevention of large-scale financial losses. Also, techniques such as supervised learning, graph neural networks, sequence modeling, and federated learning are increasingly being explored to enhance the detectability of these anomalies.

To address these potential challenges, the following sections of this report will investigate the application of AI in credit card fraud detection. This analysis will encompass a range of state-of-the-art techniques such as model architectures, machine learning algorithms and performance evaluation metrics essential for deploying a potentially more scalable and effective credit card fraud detection system for contemporary banking environments.

4.2 Overview of Machine Learning Approaches in Credit Card Fraud Detection

Credit card fraud is one of the most significant and pervasive financial crimes globally, posing a major threat to financial institutions, merchants, and consumers alike. As credit card usage increases and online transactions become more prevalent, fraudulent activities have evolved in sophistication and complexity [4]. Traditional fraud detection methods, such as rule-based and statistical approaches, are increasingly inadequate in identifying new and evolving fraud patterns in real-time.

The rise of machine learning (ML) has brought new solutions to the table, offering an opportunity to detect fraudulent transactions more accurately and efficiently. Machine learning algorithms are capable of learning from past data, identifying complex patterns, and making predictions or decisions with minimal human intervention [5]. These capabilities make them particularly effective for credit card fraud detection, where large volumes of transaction data need to be processed quickly and accurately [6].

This report explores and compares three seminal research papers on credit card fraud detection using machine learning. Each paper presents different approaches, techniques, and models for detecting fraud, as well as their associated strengths and limitations. The papers under review are:

- King, S. T., et al. "Credit Card Fraud Is a Computer Security Problem," IEEE Security & Privacy, vol. 19, no. 2, pp. 65-69, March-April 2021.
- Delamaire, Linda, Hussein Abdou, and John Pointon. "Credit card fraud and detection techniques: a review." Banks and Bank systems 4.2 (2009): 57-68.
- Omar, M. A., and D. Kiwanuka. "A State-of-the-Art Review of Machine Learning Techniques for Fraud Detection Research," in Proceedings of the 2018 International Conference on Computing and Big Data.

4.2.1 The Computer Security Problem of Credit Card Fraud

Key Findings:

King et al.'s paper explores credit card fraud detection from a computer security perspective, arguing that fraud detection is not only a financial issue but also a cybersecurity problem. The authors focus on modern machine learning techniques integrated with security protocols to enhance fraud detection.

- Fraud as a Cybersecurity Problem: The authors argue that credit card fraud should be viewed through the lens of cybersecurity. Fraudulent transactions often involve hacking, identity theft, and the manipulation of payment systems, which are akin to cyberattacks. This perspective introduces cybersecurity tools like encryption and intrusion detection systems into the fraud detection framework [1].
- Real-Time Fraud Detection: King et al. emphasize the importance of real-time fraud detection systems that can instantly flag fraudulent transactions as they occur. The paper discusses various algorithms that are suitable for real-time analysis, including anomaly detection algorithms that can spot irregularities in transaction patterns.
- Anomaly Detection Techniques: The authors delve into advanced anomaly detection techniques, which are particularly useful when fraudulent activities are unknown or do not follow known patterns. These algorithms identify unusual transactions by analyzing various features such as transaction time, location, and user behavior.
- Integration with Cybersecurity: The paper presents a model that integrates machine learning with cybersecurity tools, such as intrusion detection systems, to improve fraud detection. This combined approach allows for better identification of fraud that may involve hacking or account takeover.
- The authors propose the use of machine learning models like support vector machines (SVMs) and deep learning networks in tandem with cybersecurity systems to form a robust fraud detection system.
- They also highlight the need for continuous adaptation of models to new threats. As fraud tactics evolve, real-time detection systems must be updated with new data to remain effective.

Performance:

King et al. argue that modern fraud detection systems based on machine learning can achieve high detection rates while maintaining low false-positive rates, which is crucial for real-time fraud detection.

Limitations:

- A key limitation discussed is the computational complexity of implementing real-time fraud detection, particularly in systems with large volumes of transactions.
- Another limitation is the difficulty of obtaining labeled data for supervised learning models, as fraudulent transactions are relatively rare.

4.2.2 Credit Card Fraud Detection Techniques

Key Findings:

Delamaire et al.'s paper is a comprehensive review of credit card fraud detection methods. It highlights both traditional approaches and emerging machine learning techniques. The paper offers a valuable foundation for understanding the evolution of fraud detection technologies.

- Delamaire et al.'s paper is a comprehensive review of credit card fraud detection methods. It highlights both traditional approaches and emerging machine learning techniques. The paper offers a valuable foundation for understanding the evolution of fraud detection technologies.
- Traditional Techniques vs. Machine Learning: The paper contrasts traditional fraud detection methods, such as rule-based systems and statistical techniques, with machine learning models. It argues that while traditional methods have been effective to some degree, they often fail to detect new types of fraud and require manual rule creation and maintenance [2]. Machine learning algorithms, on the other hand, can adapt to new fraud patterns without explicit reprogramming.
- Supervised Learning Models: The authors highlight supervised learning models, particularly decision trees and neural networks, as highly effective for credit card fraud detection. These models can be trained on historical transaction data to classify transactions

as legitimate or fraudulent based on features such as transaction amount, merchant, and cardholder behavior.

- Unsupervised Learning Models: The paper also discusses unsupervised techniques like clustering and anomaly detection, which can be used when labeled data is scarce. These models are designed to identify outliers or abnormal behavior, which may indicate fraudulent activity.
- Data Imbalance Problem: One of the key findings is the issue of class imbalance, where fraudulent transactions make up a tiny fraction of the total data. This imbalance leads to high false-negative rates in many models, as the algorithm may favor predicting the majority class (legitimate transactions) at the expense of the minority class (fraudulent transactions).
- The paper emphasizes the importance of feature engineering and selection in building robust models. Feature selection techniques like principal component analysis (PCA) are crucial for reducing dimensionality and improving model performance.
- Ensemble methods like boosting and bagging are also highlighted as effective in improving the performance of fraud detection models by combining the strengths of multiple models.

Performance:

The performance of machine learning algorithms like decision trees and ANNs is shown to be superior to traditional rule-based approaches in terms of accuracy and fraud detection rates.

Limitations:

- One limitation highlighted is the challenge of handling imbalanced datasets, where fraudulent transactions are much fewer than legitimate ones.
- The need for continuous model updating and retraining to adapt to emerging fraud techniques is another limitation.

4.2.3 Machines Learning Techniques

Key Findings:

Omar and Kiwanuka's paper provides a state-of-the-art review of machine learning techniques specifically applied to fraud detection. They focus on advanced machine learning models and algorithms that show promise in improving detection accuracy.

- Supervised Learning Algorithms: The paper reviews commonly used supervised learning algorithms for fraud detection, including decision trees, random forests, and support vector machines (SVMs). These models are praised for their ability to classify transactions based on labeled historical data.
- Unsupervised Learning Algorithms: In addition to supervised learning, Omar and Kiwanuka discuss the application of unsupervised learning methods such as clustering, k-nearest neighbors (K-NN), and isolation forests. These techniques are valuable when labeled data is limited, as they can detect outliers and anomalies in transaction data.
- Hybrid Approaches: One of the most significant contributions of this paper is the discussion of hybrid models that combine multiple machine learning techniques [3]. For instance, integrating decision trees with neural networks or anomaly detection models can increase the robustness of the fraud detection system.
- Feature Engineering and Selection: The paper underscores the importance of feature engineering in improving the performance of machine learning models. The authors highlight the need to extract relevant features from raw transaction data, such as user transaction history, time of day, merchant type, and transaction location.
- The authors also discuss the limitations of current fraud detection systems, particularly in terms of scalability and adaptability. As transaction data grows, traditional machine learning models may struggle to maintain high performance without significant computational resources.
- The need for continuous retraining of models is stressed, as fraudulent activity evolves over time. Omar and Kiwanuka suggest that hybrid and ensemble models that adapt to new data patterns are particularly effective.

Performance:

The paper concludes that hybrid models, which combine supervised and unsupervised learning, tend to offer better performance in terms of detection accuracy and generalization to new types of fraud.

Limitations:

- One limitation is the challenge of selecting the right features from transaction data, as irrelevant or redundant features can degrade model performance.
- The authors also mention the scalability issues with certain machine learning models, especially when dealing with very large transaction datasets.

4.2.4 Comparative Analysis

The three papers offer valuable insights into the evolving landscape of credit card fraud detection using machine learning. Below is a comparative analysis based on the findings from the three studies.

Aspect	**King et al. (2021)**	**Delamaire et al. (2009)**	**Omar & Kiwanuka (2018)**
Focus	Fraud detection from a computer security perspective	General review of fraud detection techniques	Review of machine learning techniques for fraud detection
Machine Learning Techniques	Anomaly detection, pattern recognition	Decision trees, ANNs	Supervised learning, unsupervised learning, hybrid models
Performance	High accuracy and low false positives in real-time detection	Superiority of ML over traditional methods	Hybrid models provide the best performance
Limitations	Computational complexity, difficulty in labeled data	Data imbalance, need for model updates	Feature selection challenges, scalability issues

Table 1: Comparative analysis based on the findings

4.2.5 Models and Algorithms

- King et al. (2021): King et al. propose a more security-focused approach, highlighting the role of anomaly detection and real-time systems. Their study emphasizes that a hybrid security approach, combining machine learning with cybersecurity measures, offers the most robust solution.
- Delamaire et al. (2009): Focused on traditional machine learning models, particularly decision trees and neural networks. Their results suggest that these models can effectively identify fraudulent transactions when properly trained on large datasets.
- Omar & Kiwanuka (2018): The authors discuss various machine learning algorithms, emphasizing the importance of hybrid models. Random forests, support vector machines, and clustering techniques are discussed as effective methods for detecting fraud in different scenarios.

4.2.6 Limitations and Challenges

Each paper presents unique limitations when it comes to applying machine learning for credit card fraud detection:

- **Data Imbalance**: All three papers acknowledge the challenge of imbalanced datasets, where fraudulent transactions are significantly fewer than legitimate ones. This leads to difficulties in training models without bias.
- **Scalability**: King et al. and Omar & Kiwanuka highlight issues related to scalability, particularly when dealing with large datasets that require substantial computational resources.
- **Real-Time Detection**: Real-time detection, as emphasized by King et al., introduces computational challenges, especially in large-scale systems. The trade-off between detection accuracy and processing speed remains a critical issue.

4.2.7 Conclusion

The three papers provide valuable insights into the development and implementation of machine learning techniques for credit card fraud detection. Delamaire et al. provide a foundational overview of various detection methods, while King et al. integrate computer security techniques with machine learning for fraud prevention. Omar & Kiwanuka focus on the application of advanced machine learning algorithms, including hybrid models, to improve detection performance [7]. Despite the progress made in the field, challenges such as data imbalance, scalability, and real-time detection remain significant barriers. Further research and development are needed to address these limitations and enhance the effectiveness of machine learning-based fraud detection systems [43].

4.3 Credit Card Fraud Detection using Hidden Markov Models

4.3.1 Background of Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) based applications are common in many areas and applications such as speech recognition, bioinformatics, finance and health care [9]. More recently, the banking industry has begun utilizing Hidden Markov Models in anomaly detection for detecting possible credit card fraud [11]. This will be the focus of this section to determine if using a Hidden Markov Model approach can help improve the overall detection accuracy of possible credit card fraud.

The breakdown of this section will be as follows: a brief overview of the Hidden Markov Model (HMM) Detection Algorithm will first be described followed by a brief overview of a more simplified standalone version that was implemented for modeling the detection accuracy of this HMM Detection Algorithm. Next, the simulated output results of this more simplified standalone version will be presented followed by a brief analysis and interpretation of the results to gain further insight into how this HMM Detection Algorithm can best perform with the highest detection accuracy. Lastly, possible extensions to using Blockchain Technology for this HMM Detection Algorithm will be presented to further explore how fraud detection accuracy using HMMs can potentially be further improved through enhanced Intrusion Detection System (IDS) Data Sharing and Trust Management.

4.3.2 Hidden Markov Model (HMM) Detection Algorithm

The Hidden Markov Model (HMM) Detection Algorithm begins by initializing the HMM parameters for the hidden states (spending categories), the output observations (transaction dollar amounts), the transition probabilities (between spending categories), the output emission probabilities (between spending categories and transaction dollar amounts) and the initial state probabilities (for each spending category) [8] [9] [10]. This HMM initialization is done through a variety of training algorithms such as k-means clustering and the forward/backward procedure [9] [10]. Once the training phase of the HMM Detection Algorithm completes, we get a complete HMM that captures the spending profile and tendencies of a particular credit card user [9].

Once we have the complete HMM, the detection phase of the algorithm then kicks off to start detecting if any new incoming transactions potentially deviate from this trained HMM. The detection phase begins by first gathering together a chain of transactions of length R (usually between 10-20 transactions) used in the training phase and computing using the forward algorithm the probability that this previously observed chain of transactions from the training data is accepted by the trained HMM [9]. Since the probability of acceptance should be pretty high since these transactions were previously used in the training phase, a second new chain of transactions of the same length, but with a new incoming data transaction replacing one of the old transactions is subsequently computed using the forward algorithm to determine if the probability of the new chain drops by a significant threshold [8] [9]. If so, then the new incoming data transaction may either be an anomaly that needs to be reconciled into the spending profile of the credit card user or could be an indicator of potential credit card fraud. To determine if the new chain could be an indicator of potential credit card fraud, the drop threshold for the new chain should exceed (P1-P2)/P1 where P1 is the probability of the first chain of R transactions from the training data and P2 is the probability of the second chain of R-1 transactions from the training data plus the new incoming transaction being tested for. If the drop threshold is exceeded, then the new incoming transaction is flagged as potentially fraudulent and dropped from the chain replaced by the next incoming transaction to be tested, else the new incoming transaction is flagged as valid and added permanently to the end of the chain to be used as the new baseline chain for checking the next

incoming transaction [8] [9].

The drop threshold value usually ranges between a 30% to 70% (or 0.3 to 0.7) from empirical data with the lower range around 30% being more sensitive because a lower drop threshold means only a slight drop is needed to detect possible fraud [9]. This detection phase of the algorithm then repeats again for each new incoming transaction to determine if the next incoming transaction may be fraudulent or not until all incoming transactions have been observed.

4.3.3 Implemented Standalone HMM Detection Algorithm

The implemented standalone HMM Detection Algorithm is a simplified version of the original HMM Detection Algorithm described in the section above. A simplified version was implemented due to time constraints and to verify if detection accuracy suffers from removing some algorithmic complexities particular from the training phase of the algorithm. The training phase of the standalone version was reduced to just using the training data to compute the initial probability of HMM acceptance for the first chain of transactions of length R where the HMM is simply trained from initializing all the parameters of the HMM from the first chain of R transactions (for the hidden states, output observations, transition probabilities, output emission probabilities and initial state probabilities respectively). Additionally, the spending validity context used in the original HMM Detection Algorithm was assumed for each spending category in advance to help aid with the training phase of the standalone version (where high spending for spending categories 1- bills and 2- leisure AND low spending for spending category 3- electronics were initially assumed to be invalid and hence potential indicators of fraud) [8]. Also, depending on how many transactions in the training phase were observed for each spending category, the validity of the spending categories were updated accordingly where valid categories were marked as invalid if no transactions for that category were observed during the training and invalid categories were marked as valid if two or more transactions for that category were observed during the training.

For the detection phase of the standalone version, a majority of the HMM Detection

Algorithm was left in place since the detection phase of the algorithm follows for the most part a more straightforward approach where the next chain probability of R-1 transactions from the training data plus the new incoming transaction being tested for is computed using the forward algorithm to determine if the probability of this next chain drops below the probability of the first chain by a certain threshold. If so, then the new incoming transaction is flagged as potentially fraudulent and dropped from the chain replaced by the next incoming transaction to be tested, else the new incoming transaction is flagged as valid and added permanently to the end of the chain to be used as the new baseline chain for checking the next incoming transaction. This procedure again repeats until all new incoming transactions have been observed.

However, one additional thing added to the detection phase of this standalone version to help offset the reduced training phase and limited HMM training is a re-updating of the validity of the spending categories after each new incoming transaction depending on how many transactions for a certain spending category have been observed up to that point. This additional feature is important because it affects the final truth values for each spending category and hence the final overall detection accuracy of the standalone HMM algorithm since the detection accuracy measures the output of the HMM detection algorithm against the final truth values for each spending category. So if a transaction is flagged as potentially fraudulent, for example, this flag would be measured against the last updated truth value for the spending category of this transaction. If the spending category was last updated as being invalid, then the output of the HMM detection algorithm will match with the validity of the transaction's spending category increasing the final overall detection accuracy, else the output will not match with the validity of the transaction's spending category decreasing and hence hurting the final overall detection accuracy of the standalone HMM algorithm. The output detection accuracy results for various different simulated chain lengths and drop threshold values is shown next in section 4.3.4 below for this standalone HMM algorithm.

4.3.4 Simulated Results of the Standalone HMM Detection Algorithm

The output simulated detection accuracy results are shown in Table 2 below after running the standalone HMM algorithm on different chain lengths and drop thresholds:

	Chain Length	**5**	**10**	**15**	**20**	**25**
% Drop Threshold						
0.1 (10%)		17.78%	25%	11.43%	40%	48%
0.3 (30%)		28.89%	52.5%	57.14%	56.67%	52%
0.5 (50%)		57.78%	**92.5%**	77.14%	76.67%	**88%**
0.7 (70%)		57.78%	**92.5%**	**94.29%**	**93.33%**	**92%**
0.9 (90%)		**93.33%**	**92.5%**	**94.29%**	**93.33%**	**92%**

Table 2 - Detection Accuracy Percentages by Chain Length and Drop Threshold

The detection accuracy percentages shown in the table above use a drop threshold range between 10% (very sensitive) to 90% (not sensitive) and a chain length range of between 5 and 25 transactions. The chain length size in the standalone algorithm is equal to the size of the training data, so the chain length sizes above also represent the number of transactions used in the training phase. Additionally, the simulated results displayed in Table 2 above use a validity threshold of 2 validly found transactions for an initially invalid spending category to be updated as valid and a validity threshold of 1 validly found transaction for an initially valid spending category to be updated as valid again (if previously found to be invalid in the training phase due to a lack of transactions). Note the bolded percentages in the table above represent the best performing combinations simulated and hence the highest detection accuracy rates found. The analysis and interpretation of these results is presented next in section 4.3.5 below.

4.3.5 Analysis of the Simulated Results

From the data shown in Table 2 above, the detection accuracy appears to increase as the drop

threshold increases (or the sensitivity of the drop threshold decreases) and as the chain length or training sample size increases. However, for very large drop thresholds, the detection accuracy remains very high independent of the chain length. This could be because the accuracy of the standalone version depends more heavily on the sensitivity of the drop threshold with the chain length or training sample size only coming into play for very sensitive drop thresholds. Note: the detection accuracy percentages reported in Table 2 above reflect the true positive values. In other words, the detection accuracy reported is a measure of how often the standalone HMM detection algorithm correctly flagged the transaction as either valid or fraudulent. If the HMM algorithm incorrectly flagged the transaction as either fraudulent (type 1 error - false alarm) or valid (type 2 error - miss), then this would subtract from the detection accuracy percentage reported in Table 2 above.

The highest detection accuracy values shown in bold in Table 2 above outperform the highest detection accuracy values reported by A. Srivistava in paper [9] using the original HMM Detection Algorithm. This could be due to a number of factors including the size and quality of the input training data used and the fact that the detection algorithm used by A. Srivistava does not dynamically update the validity of the spending categories as new transactions come in. Additionally, the standalone HMM detection algorithm also tracks for what the highest dollar transaction seen in the training data was and sets this as a secondary threshold to use when checking for potential fraud. This high dollar threshold can be turned off if 2 or more transactions exceeding this threshold are flagged as valid (indicating a high dollar transaction is more likely to be valid than fraudulent moving forward). This high dollar threshold was also added to help offset the reduced training phase used by the standalone HMM detection algorithm. So these added validity threshold checks may be contributing to why the detection accuracy of the standalone version may be outperforming the original HMM algorithm.

4.3.6 Potential Extensions to Blockchain Technology

Given the success of the simulated results for this standalone HMM detection algorithm, there could be even more promise if the two major issues of data sharing and trust management typically

found in systems that detect for credit card fraud can be tackled as well [12]. This is where the HMM algorithm could be further enhanced to use Blockchain Technology since blockchaining can tackle both of these issues simultaneously [12].

For the data sharing issue, each transaction could be added to a blockchain instead of a chain or queue allowing for any party to access each transaction publicly and unalterably such as an auditor or banker for investigative purposes [14]. This way not only can a transaction be flagged as potentially fraudulent, but the record for that transaction can remain visible to anyone who needs to access it for verification purposes later such as to verify whether the transaction was indeed fraudulent or not.

For the trust management issue, each transaction could be added to a blockchain instead of chain or queue allowing for a majority of the interacting nodes within a credit card detection system to validate and agree upon whether the added transaction is valid or not [13]. This way not only can a transaction be flagged as potentially fraudulent, but the trust among the interacting nodes within that credit card detection system can be enhanced as well increasing the likelihood that the transaction is flagged correctly by the system.

There is no limit on the potential extensions of the HMM detection algorithm to Blockchain Technology since there exist other major issues within credit card fraud detection systems such as data privacy, trusted third parties and collaborative efficiency that may need to be tackled as well [12]. So expanding HMM detection systems to using Blockchain Technology could be a promising approach to not only further improving detection accuracy, but also enhancing the reliability of the system itself.

4.4 Graph-Based Learning Techniques in Preventing and Mitigating Unauthorized Access in Banking

Financial fraud, including transaction fraud, money laundering, and identity theft, remains a pressing challenge in the modern financial ecosystem. As digital systems become more complex and interconnected, so do the techniques used by fraudsters. Traditional fraud detection

approaches, based on rule-based systems or standard machine learning models, are often insufficient for detecting subtle, multi-hop fraudulent patterns that span across various entities. In light of this, graph-based techniques have become highly applicable in detecting fraudulent activities in the area of Banking.

4.4.1 Graph Neural Networks

Graph Neural Networks (GNNs), a deep learning technique, have emerged as a highly influential machine learning framework for modeling data characterized by graph or network structures. GNNs excel in learning graph representations by transforming tabular data into graph formats, making them an effective tool for capturing complex relationships within the data [15]. Their unique capability to model relationships and dependencies between entities makes them particularly well-suited for addressing fraud detection challenges. The tabular format of transactional data can be efficiently transformed into a graph structure by creating nodes that represent various entities, such as credit card accounts, merchants, and the transactions themselves. This approach allows for a more comprehensive analysis of relationships and interactions within the data [16]. Unlike traditional models that consider transactions or user accounts in isolation, GNNs examine the entire network of interactions, thereby capturing the intricate behaviors of users within the broader financial ecosystem.

Recent research in machine learning focused on graph analysis has been attracting significant attention due to the expressive capabilities of graphs. Graphs can model various systems across multiple disciplines, such as social networks in social sciences, financial and banking systems, physical systems in natural sciences, protein-protein interaction networks in biological sciences, and ecosystems in ecological sciences, among others. In recommender systems, users can be depicted as nodes, their similarities as edges, and the ratings they give to items as graph signals [17]. These roles can be categorized into three levels: node level, edge level and graph level.

4.4.1.1 Levels of GNNs Tasks

- **Node-Level:** GNNs play a crucial role in node classification within financial networks, identifying the label of each node by analyzing its attributes and connections. They

efficiently classify nodes to detect credit card and insurance fraud by examining transaction patterns and claimant-provider relationships within financial networks.

- **Edge-Level:** At the edge level, GNNs are utilized for tasks such as edge classification and link prediction, which are essential for predicting fraudulent transactions in the financial sector.
- **Graph-Level:** At the graph level, GNNs tackle issues related to graph classification and attribute prediction, making them valuable tools for systemic risk assessment in the financial sector [17].

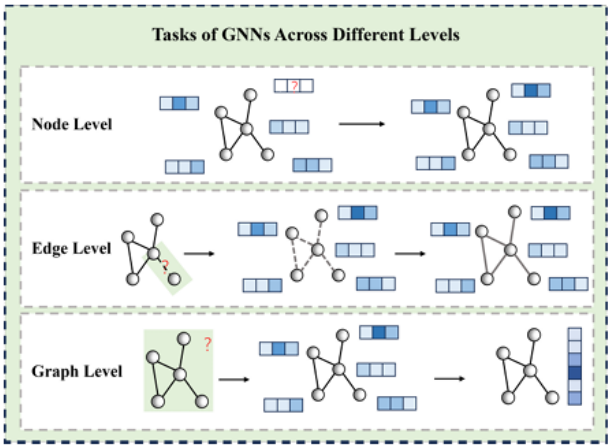

Figure 1: Different Levels of Graph Tasks [17]

4.4.1.2 Neural Message Passing

A core method for handling data in graph-based systems is Neural Message Passing. Neural Message Passing is an approach which is dependent on pairwise communication and is a defining characteristic of Graph Neural Networks (GNNs). In GNN models, messages represented as vectors are exchanged between nodes in the graph and updated using a neural network model. This process enables the model to learn representations of nodes and their relationships within the graph.

4.4.2 Graph Neural Networks Techniques

- Graph Convolutional Networks (GCN)

  Graph Convolutional Networks (GCNs) handle data organized in graph structures, efficiently detecting patterns of fraudulent activity across transactions within transaction networks. Graph Convolutional Networks (GCNs) proficiently learn features and detect patterns in financial data by capturing local connectivity and automatically identifying node and edge features. However, GCNs can experience over-smoothing, which leads to the loss of distinct node characteristics, and they are prone to overfitting when the data is limited.

- Graph Attention Networks (GAT)

  Graph Attention Networks (GATs) employ an attention mechanism, allowing the model to focus on the most relevant parts of the graph. The attention mechanism is especially beneficial in applications like financial fraud detection. Graph Attention Networks (GATs) surpass Graph Convolutional Networks (GCNs) by dynamically assigning attention weights to neighboring nodes, allowing for adaptive focus on relevant nodes and enhancing the detection of complex patterns in financial fraud.

- Graph Temporal Networks (GTN)

  Graph Temporal Networks (GTNs) use dynamic changes in financial transactions to record developing transactional data. These networks are especially successful at detecting fraud in datasets that are sensitive to temporal trends, such as credit card transactions and high-frequency trading.

- Heterogeneous Graph Neural Networks (HGNN)

  Heterogeneous Graph Neural Networks (HGNNs) are intended to manage graphs with various types of nodes and edges, allowing the model to represent a specific set of relationships and attributes in the data [17].

4.4.3 Applications of GNNs to Unauthorized Banking Access Detection

- Credit Card Fraud Detection

  Credit card fraud detection is a significant concern in the financial industry, with an emphasis on recognizing fraudulent transactions to avoid financial losses. The use of

GNNs has provided novel approaches to addressing this problem by using relational data between transactions, cardholders, and merchants.

- Online Payment Fraud Detection

  The detection of online payment fraud is crucial for protecting digital financial transactions from various sorts of fraud. GNNs, with their capacity to simulate complex, dynamic interactions between transaction entities, have emerged as a cornerstone in this field.

- Anti-Money Laundering Detection

  Anti-money laundering is a key concern in the financial sector, with the goal of identifying and prohibiting unlawful financial operations that conceal the origins of criminal gains. Transformative methods have been provided by graph neural networks, which model accounts, transactions, and entities as nodes and their relationships as edges. This approach allows GNNs to effectively detect complicated laundering patterns.

- Insurance Fraud Detection

  In the insurance industry, GNNs uncover and anticipate fraudulent trends by analyzing relational data between claims, providers, and patients. GNNs are used to detect Medicare fraud by highlighting discrepancies in provider-patient interactions [18][17].

4.4.4 Applications of Graph Attention Networks (GAT) to Unauthorized Banking Access Detection

Graph Attention Networks (GATs) stand out for their ability to learn adaptive, weighted relationships between entities in a graph, making them highly suitable for detecting sophisticated fraud patterns.

A paper [19] introduced a novel domain in their work on PU-GNN, a graph-based model for detecting chargeback fraud in Play-to-Earn (P2E) MMORPGs. The study addresses unique challenges in this context, including the irreversibility of blockchain-based in-game assets, the imbalance of labeled data (few confirmed frauds), and the presence of unlabeled but potentially fraudulent users. PU-GNN incorporates the GATv2 architecture to model token transaction networks among players. Each node (player) is represented using both behavioral embeddings derived from in-game activity logs, and contextual embeddings learned through attention-based

message passing over the transaction graph. Crucially, the attention mechanism allows the model to distinguish between benign and suspicious connections, such as identifying clusters of players engaging in coordinated token mining and transfer patterns—common among fraudulent actors.

A noteworthy application of GATs in this domain is found in the work by [20], where the authors propose a Knowledge Graph Attention Network (KGAT) framework tailored for accounting and financial fraud detection (FAFD). The KGAT model aims to uncover high-order, knowledge-rich relational patterns within financial data by leveraging a knowledge graph-based structure and attention mechanisms. The authors identify several critical challenges in financial fraud detection including.

- Fraudsters often obscure illicit activities across multiple disclosures and reporting layers.
- Auditors may lack sufficient domain knowledge or tools to detect nuanced patterns.
- Traditional data mining approaches may be ineffective when used in isolation.

To counter these issues, the KGAT framework integrates knowledge graphs, which capture entity-attribute relationships and attention-based message propagation, enabling a more relationally aware and interpretable fraud detection process. Before training their model a t-statistic-based feature selection method was used to identify financial attributes that significantly distinguish between fraudulent and non-fraudulent entities. Features were ranked and selected based on their discriminative power across multiple folds using cross-validation. This step ensures the input of the KGAT model is informative and reduces noise, which is critical in financial datasets that often contain redundant or irrelevant variables.

Another application is the Feature Aggregation based Graph Attention framework. The authors introduced a GAT-based model that combines multi-feature aggregation and node attention to improve fraud detection performance on graph-structured financial datasets. Their architecture consists of three main modules:

- **Graph Attention Module:** This module applies standard GAT layers to compute attention coefficients between nodes. These coefficients determine the importance of neighboring

nodes in learning representations. The attention-weighted aggregation enables the model to adaptively focus on fraud-prone connections.

- **Feature Aggregation Module:** Unlike simple node-level embeddings, this model aggregates multiple feature vectors from different sources (e.g., transaction metadata, customer profiles, behavioral patterns). This ensures that the model considers a richer, more contextualized representation of each node.
- **Model Design & Integration:** The output from the attention and aggregation layers is passed to a classification layer to predict fraud probabilities. This modular design facilitates the combination of both structural relationships and semantic features.

Their GAT-based model with feature aggregation significantly improved performance over traditional GCNs and standard classifiers [21].

4.4.5 Simulation Experiments and Results

To further evaluate the effectiveness of Graph based learning techniques in preventing and mitigating unauthorized banking access, the **Feature Aggregation Based Graph Attention Model** was applied to a **creditCard_2023** dataset from Kaggle. In preparing the data, the tabular data was transformed into graph-structured data with nodes as the transaction ids and edges based on 5 nearest neighbors. This was done to assess the model's ability to accurately identify fraudulent behavior within a graph-structured environment. Focus was placed on evaluating the impact of the attention mechanism and feature aggregation strategy on detection performance as used in the paper [8]. Standard classification metrics, including precision, recall, F1-score, and AUC, were used to comprehensively measure the model's effectiveness, especially in the presence of class imbalance typically observed in fraud detection tasks.

The results clearly indicate that the Graph Attention Network model used is highly effective for fraud detection in a graph-structured setting. The model achieved an overall **accuracy** of **95%**, with per class **precision**, **recall** and **F1-score** as shown in Table 3 and Figure 2. Notably, it also recorded a high **AUC-ROC** score of **0.9941** (see Figure 3), reflecting excellent discrimination between fraudulent and non-fraudulent transactions. The strong performance, particularly on the

minority class (fraud), suggests that the model is capable of accurately identifying complex and subtle patterns of fraudulent behavior. The attention mechanism allows the model to dynamically focus on the most relevant neighboring nodes, while feature aggregation contributes to a richer and more informative representation of each entity. Overall, the GAT-based approach demonstrates strong potential for real-world fraud detection applications, particularly in domains where relational data plays a crucial role.

Class	**Precision**	**Recall**	**F1-Score**
Non-Fraud (0)	0.92	0.99	0.96
Fraud (1)	0.99	0.92	0.95

Table 3: Classification Results

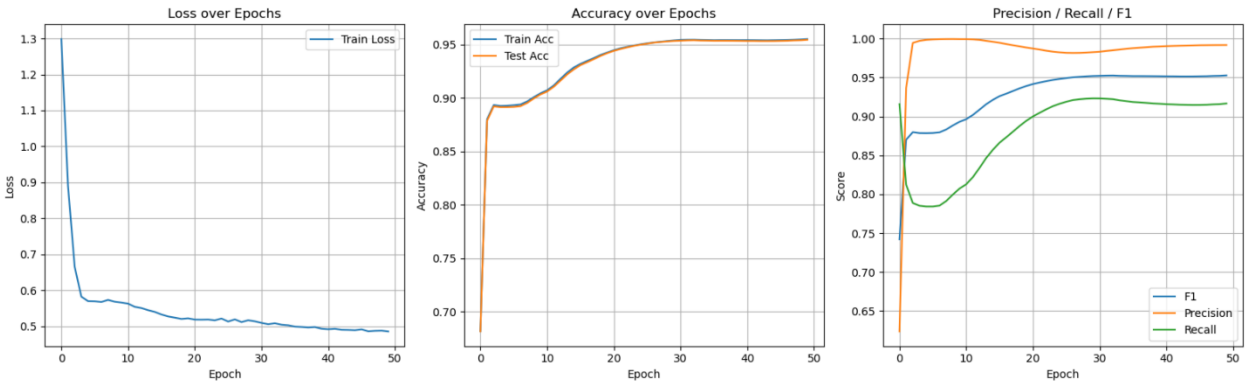

Figure 2: Loss and Evaluation Metrics over Epochs

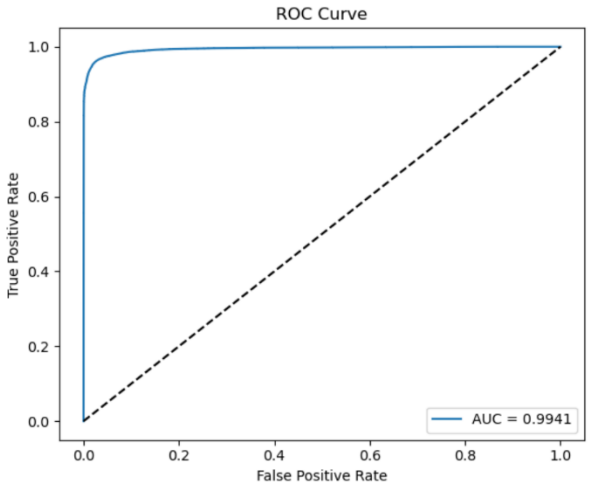

Figure 3: ROC-AUC plot

Graph Attention Network (GAT) has shown significant improvements in Unauthorized banking access detection in literature [19][20][21][44]. By leveraging attention mechanisms and feature aggregation, especially for this model, it was able to dynamically focus on relevant node relationships and enrich node representations, leading to highly accurate classification results.

These further simulation results demonstrate the model's strong capability to detect both fraudulent and non-fraudulent instances with high reliability, even in the presence of subtle and complex behavioral patterns. The effectiveness of GAT in capturing relational dependencies suggests promising applicability in real-world fraud detection systems, especially where entity interactions play a crucial role.

4.5 Credit Card Fraud Detection using Ensemble Machine Learning Models

4.5.1 Background of Ensemble Machine Learning Techniques

The dramatic surge in global financial transactions, fueled by the rise of digital banking and online shopping, has introduced an equally concerning increase in fraudulent activities. Traditional fraud detection methods, which rely on static rule-based systems, have shown significant limitations in adapting to the ever-evolving patterns of fraudsters. These systems, although effective in identifying known patterns, fail to adapt to novel fraud tactics, leading to a substantial number of false negatives.

To address this, machine learning-based approaches have been adopted by many financial institutions. Among these, ensemble learning techniques have emerged as one of the most powerful strategies. Ensemble models, by definition, combine the strengths of multiple weak learners to produce a model that is more accurate, stable, and generalizable than any single constituent model. This is achieved through techniques such as bagging and boosting. In bagging, exemplified by the Random Forest algorithm, multiple models are trained in parallel and their outputs are averaged. Boosting methods like XGBoost, LightGBM, CatBoost, and AdaBoost train models sequentially, where each model attempts to correct the errors made by its predecessor [25].

These models are particularly well-suited to fraud detection due to their capability to detect intricate patterns in high-dimensional, imbalanced datasets. Fraud detection datasets are typically characterized by an extreme imbalance between the fraudulent and legitimate transactions, often with fraud comprising less than 0.2% of all transactions. Ensemble methods handle this imbalance effectively, especially when combined with resampling strategies or cost-sensitive learning, making them a natural fit for the task [25].

4.5.2 ML Models Used for Credit Card Fraud Detection

In this study, we compared seven different machine learning algorithms, ranging from classical linear models to state-of-the-art ensemble methods. The models were trained on a benchmark dataset containing 284,807 transactions, of which only 492 were labeled as fraudulent. This provided a realistic testing ground for evaluating model robustness against class imbalance.

Logistic Regression was chosen as the baseline due to its simplicity and interpretability. Decision Trees served as a foundational model, providing insight into model behavior and feature importance. Random Forest extended the idea of Decision Trees by aggregating predictions from numerous trees to reduce variance.

The boosting algorithms included in the study—XGBoost, LightGBM, CatBoost, and AdaBoost—represent cutting-edge advancements in gradient boosting frameworks. Each has unique advantages: XGBoost introduces regularization for improved generalization; LightGBM is optimized for efficiency and handles large datasets with high speed; CatBoost is tailored for

datasets with categorical variables and combats overfitting with ordered boosting; AdaBoost iteratively focuses on misclassified samples to improve model performance [23].

4.5.3 Results and Accuracy Metrics of All Models

Each model was evaluated using classification performance metrics that are particularly relevant for imbalanced datasets: Precision, Recall, F1-score, and AUC (Area Under the Receiver Operating Characteristic Curve). The table below summarizes the performance:

Model	AUC Score	Precision	Recall	F1 Score
Logistic Regression	0.9649	0.9247	0.9240	0.9242
Decision Tree	0.9980	0.9863	0.9859	0.9861
Random Forest	0.999882	0.9986	0.9985	0.9985
XGBoost	0.999773	0.9980	0.9978	0.9979
AdaBoost	0.968600	0.9319	0.9311	0.9315
CatBoost	0.999434	0.9976	0.9972	0.9974
LightGBM	0.999223	0.9971	0.9966	0.9968

The results reveal that ensemble models significantly outperform the baseline classifiers. Random Forest achieved the highest performance with nearly perfect AUC and F1-score, indicating its strength in capturing both fraudulent and legitimate transactions. Boosting methods, especially CatBoost and XGBoost, also exhibited excellent performance, with only marginal differences in predictive capability.

4.5.4 In-Depth Analysis of the Results

The baseline models (Logistic Regression and Decision Tree) served their purpose in highlighting the comparative effectiveness of more advanced algorithms. Logistic Regression, although fast and interpretable, struggled with non-linearities and failed to match the precision of ensemble models [26]. Decision Trees performed better due to their ability to split on complex conditions but lacked the ensemble effect that provides robustness.

Random Forest, with its parallel tree training and averaging, was the top performer. It managed to minimize overfitting while maintaining high recall—a critical metric in fraud detection. Missing a fraudulent transaction (false negative) is more costly than incorrectly flagging a legitimate one (false positive), making high recall essential [22].

XGBoost and CatBoost leveraged boosting mechanisms to incrementally improve model performance. Their success stems from iterative error correction and effective use of regularization [25]. LightGBM, while slightly behind in metrics, provided significant advantages in training speed and scalability, making it ideal for real-time applications [23].

4.5.5 Contributions and Learning Outcomes

Throughout the project, I actively contributed to all aspects of the machine learning workflow. This included data preprocessing, feature engineering, model selection, training, evaluation, and visualization. I implemented multiple resampling strategies and conducted hyperparameter optimization using both grid search and randomized search. I further utilized SHAP (SHapley Additive Explanations) to explain individual predictions and understand model behavior [24].

This project allowed me to develop practical skills in handling highly imbalanced data, choosing the right evaluation metrics, and building scalable ML pipelines. It deepened my understanding of how ensemble models outperform traditional techniques and the trade-offs between accuracy, interpretability, and computational efficiency. I also explored the integration of Autoencoders for unsupervised anomaly detection in future work [28].

4.5.6 Future Work and Recommendations

While the current results are promising, there is substantial scope for improving the system further by addressing the limitations of static models and enhancing their adaptability in real-world environments. One critical area is the deployment of real-time fraud detection systems. Traditional batch processing is often inadequate in fast-paced financial environments where latency can result in significant losses. Streaming architectures leveraging Apache Kafka or Apache Flink can be introduced to continuously ingest and process transactions. Models can be updated incrementally using online learning techniques to reflect the most recent fraud patterns [23].

Additionally, exploring advanced neural network architectures such as Long Short-Term Memory (LSTM) and Transformer models could significantly improve temporal pattern recognition, especially in sequential data streams. These models are capable of learning dependencies across time steps, which is essential in capturing the behavior of fraudsters who often act in bursts or follow hidden time-based patterns [27].

Another promising direction is the use of hybrid models that integrate unsupervised anomaly detection with supervised learning. Unsupervised methods like Isolation Forests or Autoencoders can help detect novel fraud patterns that the supervised models have never seen [28]. These hybrid frameworks could be particularly useful when dealing with concept drift, where the statistical properties of incoming data change over time [24].

To improve recall further without sacrificing precision, synthetic data generation methods like SMOTE (Synthetic Minority Over-sampling Technique) or generative adversarial networks (GANs) can be used to balance the dataset and enhance model training [22]. The quality of synthetic data and its impact on performance can be evaluated using adversarial validation techniques.

Moreover, the integration of interpretability tools like LIME and SHAP in production systems will be essential for gaining regulatory approval and building trust among stakeholders. Especially in the finance sector, explainability is not optional but mandatory under guidelines such as the General Data Protection Regulation (GDPR).

Finally, a formal deployment strategy involving RESTful API interfaces, containerization (Docker), and orchestration platforms (Kubernetes) can be adopted to move the system into production. Monitoring systems should be integrated to log predictions, track model drift, and trigger retraining when necessary [25].

4.6 Comparative Analysis of Machine Learning and Deep Learning Methods

This section provides a comparative analysis of three prominent AI-based approaches to detect and mitigate unauthorized access in banking systems, based on insights from all seven reviewed papers: [29], [30], [31], [32], [33], [34], and [35]. The analysis emphasizes the differences in methodology and effectiveness for integration into contemporary banking environments. Techniques that will be considered include ensemble machine learning models and time-series behavior profiling. This comparative lens ensures that the technical landscape is not only presented but also critically evaluated within the scope of banking security.

4.6.1 Supervised Learning for Transaction Classification

4.6.1.1 Random Forest: Overview

Supervised learning refers to a branch of machine learning in which models are trained using labeled datasets, enabling them to learn the relationship between input features and corresponding output labels. In the context of fraud detection, these models are used to classify transactions as either fraudulent or legitimate based on patterns learned from historical data. Among the widely used supervised learning approaches is the Random Forest algorithm—an ensemble method that constructs multiple decision trees and aggregates their predictions. This technique helps improve classification accuracy while minimizing the risk of overfitting and offers greater reliability compared to a single decision tree model.

4.6.1.2 Methodology and Implementation

Supervised learning models learn patterns from labeled datasets to classify new inputs. In fraud detection, they are trained to distinguish between fraudulent and legitimate transactions based on

historical data. Random Forest, as discussed in [29] and [31], is a supervised ensemble technique that constructs multiple decision trees on randomly sampled feature subsets and datasets. Xuan et al. investigated two variants:

- **Random Forest I:** Leveraging randomly generated trees.
- **Random Forest II:** Built using structured Classification and Regression Trees (CART) guided by Gini impurity.

Training was conducted on transaction datasets with highly skewed fraud-to-legitimate ratios, ranging from 1:1 to 10:1. Class balancing was applied using under-sampling techniques.

4.6.1.3 Performance and Limitations

Random Forest II consistently outperformed Random Forest I across recall, F1-score, and precision. It demonstrated stability under class imbalance, crucial in real-world banking where fraud is rare. However, excessive tree depth reduced interpretability, and training time scaled with data volume.

4.6.1.4 Model Comparison

4.6.1.4.1 Decision Tree

The Decision Tree model creates a hierarchical tree-like structure where nodes represent features and branches represent decision rules. In the study, it demonstrated high interpretability and rapid prediction time (0 seconds for testing). Sensitivity remained stable at 79.21% across threshold changes (0.5 and 0.4), while precision stood at 85.11%. Due to its balance between efficiency and moderate predictive power, it was recommended by the authors as the most practical choice for real-time fraud detection despite kNN offering slightly better sensitivity.

4.6.1.4.2 k-Nearest Neighbors (k-NN)

k-NN uses the majority label of the 'k' nearest neighbors, measured by distance metrics such as Euclidean or Minkowski. While kNN achieved the highest sensitivity at 81.19% and a strong

precision of 91.11%, its practicality was limited by a test prediction time of 462 seconds. Despite its effectiveness, its computational cost makes it less suitable for high-frequency banking environments.

4.6.1.4.3 Naive Bayes

This probabilistic model applies Bayes' Theorem under the assumption that features are independent. It was the fastest model with 0 seconds for both training and testing. Sensitivity was highest at 85.15%, but precision was only 6.56%, indicating a high false-positive rate. Its assumption violations severely impacted classification reliability.

4.6.1.4.4 Logistic Regression

Logistic Regression, a linear model used for binary classification, improved with threshold adjustment (sensitivity rose from 63.34% to 69.31%, precision remained around 87.5%). It required minimal training and prediction time. It performed adequately but could not model complex fraud relationships effectively due to its linear nature.

4.6.1.4.5 Random Forest

As an ensemble of decision trees, Random Forest showed strong results with precision increasing to 93.83% and sensitivity at 78.22% (threshold 0.4). Its training time (23 seconds) was longer than simpler models, but testing time remained negligible. It provided the best trade-off between precision, sensitivity, and computational feasibility. Compared to other methods, Random Forest provides a balanced trade-off between accuracy and interpretability. It is better suited for transaction-level fraud detection than for understanding behavioral sequences or high-dimensional data.

4.6.1.5 Deep Learning Ensembles

Recent work has focused on improving credit card fraud detection using deep learning ensembles with data resampling. For example, Mienye and Sun proposed an LSTM-based ensemble using AdaBoost and SMOTE-ENN data resampling. This method achieved 0.996 sensitivity, 0.998

specificity, and 0.990 AUC, outperforming traditional classifiers like SVM, Multilayer Perceptron, and even standalone LSTMs [34].

Similarly, Esenogho et al. utilized LSTM as a base learner in an AdaBoost ensemble. Results showed that their ensemble outperformed traditional methods and benefited significantly from hybrid resampling [33]. These approaches effectively handled class imbalance and temporal dependencies in transaction sequences.

4.6.2 Time-Series Behavioral Profiling

4.6.2.1 Overview

Time-series modeling involves analyzing sequential patterns of user behavior over time. In fraud detection, this technique enables systems to establish a behavioral baseline for each user and detect deviations that may indicate unauthorized access. Time-series models are especially effective in identifying fraud that mimics legitimate transactions in amount but differs in timing or frequency.

4.6.2.2 Methodology and Implementation

Time-series models focus on the temporal behavior of users by analyzing sequences of transaction patterns over fixed intervals. This allows for behavioral profiling and anomaly detection over time. Seyedhossein and Hashemi [30] proposed a time-series based system that aggregates user transactions into 7-day windows. Profiles were clustered using:

- **Euclidean Distance:** Captures similarity in transaction volume.
- **Permutation-Aligned Euclidean Distance (PAED):** Accounts for shifts in transaction timing without penalizing behavioral consistency.

Profiles were categorized into stable, permuted, or unpredictable, and thresholds (θ1–θ3) were used to assess anomaly severity.

4.6.2.3 Performance and Limitations

The model excelled in detecting fraud among regular users, identifying deviations invisible to transaction-focused classifiers. However, for unpredictable users or those with limited historical data, detection latency and reliability have diminished.

4.6.2.4 Comparative Remarks

This approach is unique in its behavioral lens, making it ideal for fraud that imitates transaction values but disrupts habitual sequences. However, it lacks the precision of classifier-based models in individual transaction classification.

4.6.3 Class Imbalance and Evaluation

4.6.3.1 Rebalancing Techniques

Class imbalance is a frequent issue in fraud datasets where fraudulent samples are vastly outnumbered by legitimate ones. Various strategies are used to rebalance the data to improve detection accuracy and reduce bias toward the majority class. The techniques employed were:

- **Random Under-sampling:** Used in [29][31] to equalize class distribution.
- **Cost-sensitive Learning:** Adjusts model penalties for misclassified fraud.
- **SMOTE-ENN:** A hybrid oversampling and cleaning method used in [33], [34], and [35] to synthesize minority class samples and remove borderline examples.

4.6.3.2 Evaluation Metrics

Traditional accuracy was deemed insufficient. Metrics like recall, precision, F1-score, and AUC-PR were emphasized across all papers.

4.6.3.3 Comparative Remarks

Time-series models suffered less from imbalance due to their structural flexibility, while supervised classifiers required explicit sampling strategies.

4.6.4 Overall Insights and Comparative Reflections on AI Techniques

- These seven research studies contributed significantly to the exploration of AI-driven solutions for unauthorized access detection in financial systems. They emphasized trade-offs among precision, latency, robustness, and adaptability.
- Deep learning ensembles such as LSTM-AdaBoost with SMOTE-ENN delivered superior performance metrics and addressed sequence modeling challenges [33], [34].
- GRU-based architectures also emerged as powerful contenders due to their ability to efficiently model temporal dependencies with fewer parameters [32].
- Mienye and Jere provided a broader review of DL algorithms such as CNNs, LSTMs, GRUs, and Transformers, noting that GRUs achieved the best trade-off in sensitivity and F1-score, while Transformer-based models showed high accuracy [32].
- Pozzolo et al. addressed real-world fraud detection challenges such as concept drift and verification latency. Their proposed learning strategy used separate classifiers for delayed and feedback-supervised samples, improving fraud detection in live banking environments [35].
- The collected evidence strongly advocates for hybrid approaches that blend deep learning, ensemble learning, time-series analysis, and resampling. This hybrid defense mechanism ensures both precision and robustness in detecting unauthorized banking access.

4.7 Deep Learning

Deep learning has garnered significant attention in recent years, particularly with the advent of transformative applications such as ChatGPT, which exemplify the potential of deep neural architectures. Unlike traditional neural networks, deep learning models are composed of multiple layers that iteratively process and refine input data. Each layer is designed to capture increasingly abstract features, enabling the model to distinguish relevant patterns and make complex inferences. These models are trained using large batches of data, with optimization techniques such as stochastic gradient descent employed to adjust the weights of each layer. The hierarchical and distributed nature of deep learning allows it to generalize effectively, filtering out noise and focusing on salient data attributes.

Deep learning has been successfully applied in various domains, including image recognition, genomic mutation prediction, and natural language processing tasks such as topic classification [36]. In the context of credit card fraud detection, deep learning provides a powerful mechanism to model consumer behavior and identify anomalies. Our approach leverages this capability by training models to understand the structure and usage patterns of credit card transactions.

The initial phase of training involves exposing the model to a dataset comprising both legitimate and fraudulent credit card transactions. This step is essential for the model to learn the semantics and expected structure of transaction inputs. Once the model has acquired this foundational knowledge, it will be trained to assess transaction sequences—taking into account the consumer's historical spending behavior over an extended period (e.g., two years)—to determine the likelihood that a current transaction is fraudulent [37]. This behavior-aware design allows the system to make more context-sensitive decisions, reducing false positives by recognizing legitimate but unusual purchases. However, there remains a risk of rejecting genuine transactions if they deviate significantly from prior patterns. To mitigate this, we ensure that models are trained on sufficiently long consumer histories, providing a robust basis for behavioral modeling. There is a security problem that might arise from this. Because the model learns credit cards if a person were to use the model to create or identify credit card information that would go against the purpose, careful procedures must be made that the model can't identify exact credit cards and instead only understand whether the inputted information makes sense for a credit card purchase.

4.7.1 Federated Learning

One of the principal challenges associated with deep learning is the requirement for large-scale, high-quality datasets to train models that generalize effectively to real-world applications. While publicly available datasets offer a starting point, they often lack the diversity and volume necessary for robust performance. In the context of credit card fraud detection, data availability is further restricted due to stringent privacy regulations and the isolated nature of institutional data silos. Credit card transaction data is typically held exclusively by the issuing financial institution, making centralized data aggregation impractical or non-compliant with privacy standards such as GDPR and PCI DSS.

To overcome these limitations, this work adopts federated learning (FL)—a decentralized machine learning paradigm that enables collaborative model development across multiple clients without requiring the exchange of raw data. In the FL framework, each client (e.g., an individual bank) trains a local model using its proprietary dataset. The locally trained model updates are then aggregated into a global model by a central server or coordination mechanism. This approach ensures that sensitive transaction data remains within the originating institution, thereby preserving privacy while enabling knowledge sharing across participants. The aggregated global model benefits from a more diverse dataset, potentially improving accuracy and generalizability in fraud detection tasks.

Federated learning can be implemented through various architectures, depending on the distribution of data across clients [38]. These include Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL):

- Vertical Federated Learning (VFL) is applicable when multiple organizations possess datasets that share the same user base but differ in feature space. For example, a bank and an e-commerce company might both have information about the same customer but with non-overlapping features. This configuration is not particularly relevant for our scenario, as it assumes shared users and distinct feature sets—conditions that are unlikely to hold across banks, which generally collect similar types of credit card transaction data and serve different customer populations.
- Horizontal Federated Learning (HFL) is the most suitable approach for this study. HFL is employed when clients share a common feature space but differ in sample space—that is, each bank holds data on distinct sets of customers but collects similar types of features (e.g., transaction amount, location, merchant category). While a minor overlap in customers may exist if individuals hold accounts at multiple institutions, such cases are exceptions and do not undermine the general applicability of HFL in this context.
- Federated Transfer Learning (FTL) is designed for scenarios where both the sample space and feature space differ across clients, with only limited overlap. FTL extends a pre-trained

model with new features or data domains. This approach is more appropriate in cross-domain applications and is less relevant for our credit card fraud detection task.

Based on the nature of credit card transaction data, Horizontal Federated Learning emerges as the most appropriate framework. It enables secure, privacy-preserving collaboration among banks, facilitating the development of a robust and scalable fraud detection model capable of adapting to diverse and evolving transaction behaviors.

4.7.2 Feedforward Neural Networks

Feedforward Neural Networks (FNNs) are among the most fundamental and widely studied architectures within the field of deep learning. These networks consist of an input layer, one or more hidden layers, and an output layer, where each neuron in a given layer is connected to all neurons in the subsequent layer. The core mechanism involves computing a weighted sum of inputs, applying a nonlinear activation function, and propagating the result forward through the network. During training, the model utilizes backpropagation to iteratively adjust weights in order to minimize a predefined loss function, thereby improving prediction accuracy [39].

FNNs have been successfully applied to a variety of classification and regression tasks, however, despite their versatility, FNNs exhibit significant limitations when applied to tasks involving sequential data. Specifically, FNNs operate under the assumption that all inputs are independent of one another, lacking any internal memory mechanism to retain context from previous inputs. In the domain of credit card fraud detection, this constraint is particularly problematic. Fraudulent activity often manifests as outliers in a sequence of transactions rather than in isolated purchases. Consequently, a model that is unaware of prior transaction history may fail to detect fraudulent behavior that only becomes apparent over time.

Due to this inability to model temporal dependencies, FNNs are not well-suited for applications where the detection of anomalies depends on the context and order of inputs. As such, while FNNs may offer reasonable performance in scenarios using aggregated or engineered features, their lack

of sequential modeling capabilities renders them suboptimal for our proposed framework, which emphasizes the temporal dynamics of individual credit card usage patterns.

4.7.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) represent a class of artificial neural networks specifically designed for processing sequential data. Unlike feedforward neural networks, which assume independence between inputs, RNNs utilize recurrent connections to retain information across time steps. This enables them to maintain a form of memory by updating a hidden state that captures information from previous inputs in the sequence. As a result, RNNs are well-suited for modeling temporal dynamics and dependencies, making them particularly effective in domains such as language modeling, speech recognition, and financial transaction analysis [40].

Traditional models used in sequence-based tasks, such as n-gram models or fixed-window classifiers, often struggle to capture long-term dependencies due to their limited context window. RNNs overcome this by introducing an architecture where the hidden state at time step t is a function not only of the input at t, but also of the hidden state from $t-1$. This design allows the network to, in principle, learn patterns and relationships that span over long sequences of data.

In the context of credit card fraud detection, the sequential nature of transaction histories makes RNNs a natural fit. Each transaction can be treated as a time step, enabling the model to detect anomalies or patterns in consumer spending behavior over time. While a known limitation of standard RNNs is their difficulty in capturing long-term dependencies due to the vanishing and exploding gradient problem during training, this challenge is mitigated in our case. Since our input sequences (i.e., a fixed number of previous transactions) are bounded and relatively short, the issue of learning dependencies across very long sequences is minimized.

Nevertheless, to further enhance the model's capacity to retain and manage temporal information, especially when larger input windows are required, techniques such as gradient clipping can be applied to stabilize training. Moreover, advanced gated architectures like Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs) can be employed. These

architectures introduce gating mechanisms that regulate the flow of information, allowing the network to more effectively preserve relevant features over time while discarding less useful data. These architectures consistently outperform traditional models in tasks that require a deep understanding of sequential context, making them highly suitable for fraud detection systems that need to interpret nuanced transaction patterns [40].

4.7.4 Transformers

Transformers have emerged as a state-of-the-art architecture in deep learning, initially designed for natural language processing tasks but now widely adopted across domains including vision, speech, and structured tabular data. The fundamental innovation of the Transformer architecture lies in its self-attention mechanism, which enables the model to weigh the relative importance of each element in an input sequence. Unlike recurrent neural networks (RNNs), which process sequences sequentially, Transformers operate on the entire input sequence in parallel. This architectural shift allows for significantly improved scalability, faster training, and the ability to capture long-range dependencies more effectively [41].

In the context of credit card fraud detection, Transformers offer a promising framework for modeling sequences of financial transactions. Each transaction is encoded as a fixed-length vector capturing features such as transaction amount, merchant category, location, timestamp, and user ID. To preserve temporal ordering—an essential aspect of behavioral modeling—positional encodings are added to the input vectors before being passed through the Transformer layers. The multi-head self-attention mechanism then enables the model to evaluate the relevance of a current transaction with respect to the entire history of prior transactions, allowing for context-rich and behavior-aware decision-making.

Despite their strengths, Transformers present notable challenges. They are computationally demanding, often requiring specialized hardware such as GPUs or TPUs, and are sensitive to hyperparameter selection (e.g., number of attention heads, layer depth, learning rate). Furthermore, Transformers tend to perform best when trained on large-scale datasets. Given the imbalanced nature of credit card fraud—where fraudulent transactions are relatively rare—the model may risk

underfitting if not adequately regularized. Techniques such as dropout, early stopping, and data augmentation may help mitigate these risks, but the inherent data sparsity remains a challenge when applying Transformers to fraud detection tasks.

4.7.5 BERT

Bidirectional Encoder Representations from Transformers (BERT) represents a significant evolution of the Transformer architecture, particularly in its use of masked language modeling (MLM) as a pre-training strategy. Unlike conventional Transformers that process inputs in a unidirectional or fixed causal manner, BERT is trained to predict masked tokens within a sequence using context from both past and future positions, thereby enabling a deep bidirectional understanding of input data [42].

This pretraining approach can be adapted for structured data domains, including fraud detection. For example, BERT can be pre-trained on large volumes of unlabeled credit card transaction data by randomly masking certain input features—such as the transaction amount, merchant code, or timestamp—and training the model to predict the masked values. Through this process, BERT learns structural regularities and latent relationships in consumer spending behavior, which can later be fine-tuned for fraud detection tasks using labeled data.

The bidirectional nature of BERT is particularly advantageous in credit card fraud detection, where the legitimacy of a transaction is often context-dependent. A high-value purchase, for instance, may appear anomalous in isolation but could be part of a recurring pattern (e.g., monthly utility payments). BERT's ability to evaluate a transaction in the context of both preceding and succeeding events enables it to make more nuanced and context-aware classifications.

While BERT requires substantial computational resources during pretraining, it offers a flexible and powerful foundation for downstream tasks. Pretrained domain-specific variants, such as FinBERT, or lightweight derivatives like DistilBERT, can be employed to reduce computational overhead while maintaining performance. These adaptations make BERT a highly promising

model for real-time fraud detection systems that must reason over sequential transaction data with precision and context sensitivity.

5. Conclusions and Recommendations

5.1 Conclusions

This project explored the critical issue of unauthorized access in banking through the lens of artificial intelligence (AI), implementing and analyzing techniques across supervised learning, graph-based modeling, temporal behavior profiling, deep learning architectures, federated systems, and HMM anomaly detection. Each method was assessed based on its practical applicability, detection accuracy, and adaptability to changing patterns of fraudulent activity.

Supervised learning algorithms—including Random Forests, Decision Trees, Logistic Regression, Support Vector Machines (SVMs), and various boosting frameworks—were evaluated across multiple experimental configurations. These models, while relatively easy to implement and interpret, were found to be unlike anomaly detection limited in adapting to previously unseen fraud strategies, underscoring the need for more adaptive solutions.

Ensemble models demonstrated strong predictive performance, particularly under class-imbalanced conditions prevalent in fraud datasets. Algorithms such as Random Forest, XGBoost, CatBoost, and LightGBM consistently outperformed single models in F1-score and recall. The ability of these methods to combine the outputs of multiple base learners proved beneficial in minimizing false negatives and improving model robustness.

Temporal modeling and behavioral profiling provided a dynamic layer of analysis by capturing user-specific patterns across sequences of transactions. Techniques like sliding window analysis and distance-based behavioral metrics enabled early detection of suspicious behavior, even when transaction amounts or categories did not appear inherently abnormal.

Deep learning approaches, particularly Recurrent Neural Networks (RNNs), Transformer-based architectures, and BERT-style encoders, added substantial depth to the modeling of sequential transaction data. These models proved adept at capturing nuanced temporal dependencies and contextual shifts. Additionally, the use of attention mechanisms improved interpretability and

allowed models to assign greater significance to anomalous behavior patterns within transaction streams.

Hidden Markov Models (HMMs) offered a lightweight, flexible approach for detecting sequential anomalies in transaction behavior. Their probabilistic structure allowed for real-time classification based on user learned behavioral spending profiles. Enhancements such as adaptive drop thresholds and spending category validity updates also extended their usefulness to real-time banking scenarios.

Graph-based learning emerged as a particularly powerful tool for identifying complex fraud structures embedded in relational data. Models such as Graph Neural Networks (GNNs), Graph Convolutional Networks (GCNs), and Graph Attention Networks (GATs) effectively modeled relationships between customers, merchants, and transactions. The deployment of a custom GAT model yielded excellent classification results and confirmed the importance of modeling inter-entity interactions in banking fraud detection.

Federated Learning (FL), specifically the Horizontal Federated Learning (HFL) paradigm, was identified as a compelling privacy-preserving strategy. HFL enables model training across multiple institutions without exchanging sensitive customer data. This architecture supports regulatory compliance while promoting shared intelligence among institutions.

The importance of incremental evaluation and validation workflows was also emphasized. Designing test environments that simulate real-world constraints—such as microservice-based deployments and delayed-label evaluation—proved essential for stress-testing model performance before production deployment.

Operational challenges such as latency, scalability, class imbalance, and model explainability were acknowledged as central considerations in deploying AI for fraud detection. Explainable AI (XAI) techniques were recognized as essential for ensuring interpretability, especially in high-stakes environments like banking where trust and transparency are mandatory.

In conclusion, the findings of this project strongly suggest that certain AI methods such as supervised, unsupervised, sequence-based, and graph-based methods potentially offer more accurate and scalable solutions while other methods like anomaly detection potentially offer more adaptive solutions to the complex and ever-evolving landscape of unauthorized access in banking.

5.2 Recommendations

Implement Multi-Tier Fraud Detection Architectures:

Financial institutions are encouraged to design layered AI systems that integrate transaction-level classifiers, behavioral modeling, and graph-based reasoning. This multi-tier approach can detect fraud across different abstraction levels—from individual transaction anomalies to network-based fraud rings—offering comprehensive system protection.

Integrate Federated Learning for Privacy-Preserving Collaboration:

Institutions should consider implementing Horizontal Federated Learning to facilitate secure, collaborative model development across banks. This strategy enables decentralized learning without sharing raw data, thereby maintaining compliance with data privacy regulations such as GDPR and PCI DSS.

Expand the Use of Temporal and Behavioral Profiling:

Models capable of modeling long-term user behavior, such as RNNs, Transformers, and HMMs, should be central to fraud detection pipelines. These models can detect gradual shifts in user behavior, identify account takeovers, and expose time-sensitive fraud patterns often missed by static models.

Prioritize Model Explainability for Regulatory Compliance:

Integrating interpretability frameworks such as SHAP, LIME, and attention-based visualizations is essential for meeting regulatory standards and building trust with stakeholders. Transparent models also facilitate debugging, internal audits, and ongoing optimization.

Adopt Real-Time Stream Processing Architectures:

Fraud detection systems should be designed with real-time capabilities using tools such as Apache Kafka, Flink, or Spark Streaming. These platforms enable immediate fraud detection, reducing response time and minimizing financial losses from delayed interventions.

Ensure Continuous Model Adaptation and Drift Monitoring:

To address the non-stationarity of fraud behavior, institutions should implement adaptive learning mechanisms. These include active learning, concept drift detection, and periodic retraining based on feedback from human analysts and system alerts.

Promote Graph-Based Learning and Visual Fraud Analysis:

Investment in graph-based models such as GATs and GCNs is recommended to uncover complex interconnections between entities. Visualization tools should be used alongside these models to provide analysts with intuitive fraud exploration interfaces.

Balance Detection Performance Using Cost-Sensitive Optimization:

Fraud detection models should be tuned not only for overall accuracy but also for the cost trade-offs between false positives and false negatives. Cost-sensitive learning, threshold tuning, and evaluation based on recall and precision metrics are necessary for optimal performance.

Incorporate Domain Expertise into Feature Engineering:

Expert knowledge about financial operations, such as transaction categories and merchant behavior, should be incorporated during feature design. Carefully engineered features can dramatically improve model performance, particularly when labeled data is limited.

Facilitate Cross-Institutional Fraud Intelligence Sharing:

Banks and financial regulators should support cross-border collaboration through anonymized pattern sharing and federated AI platforms. This collaborative approach will enhance early detection of emerging fraud strategies that span multiple jurisdictions.

Design Systems for Modularity and Scalability:

AI systems should follow microservice architectures and be deployed using containerized tools like Docker and Kubernetes. This ensures flexible scaling, faster updates, and seamless integration with legacy infrastructure.

Leverage Synthetic Data for Model Training and Benchmarking:

The use of synthetic data generation tools such as SMOTE, GANs, or CTGAN should be promoted for benchmarking and augmenting training data. Such datasets are crucial for testing models under edge-case scenarios without compromising data privacy.

Expand Use of Transfer Learning and Pretrained Models:

Transfer learning should be used to accelerate model development in data-sparse environments. Pretrained models such as BERT-style encoders can be fine-tuned for fraud detection tasks, improving performance with minimal additional training.

Establish AI Governance and Oversight Frameworks:

Banks should implement governance protocols that oversee the entire AI model lifecycle—from development to deployment and auditing. This includes regular fairness assessments, bias mitigation practices, and transparent reporting mechanisms aligned with ethical AI standards.

Incorporate SHAP-Based Interpretability for Post-Hoc Analysis

SHAP (SHapley Additive Explanations) should be used not only for real-time model interpretation but also for analyzing model behavior across fraud cases. This supports fairness audits and helps identify hidden biases in the system.

Utilize Multi-Feature Aggregation in Graph-Based Models

Graph models should combine behavioral, transactional, and demographic features into unified node representations. This improves the richness of learned embeddings and enhances fraud detection accuracy.

Validate Graph Attention Networks Using Realistic Graph Transformations

GAT-based fraud detection models should be validated on graph-structured data derived from real transaction logs. Constructing graphs using transaction IDs and nearest neighbors enables effective, context-aware fraud modeling.

Apply Feature Selection Before Graph Modeling to Improve Learning Efficiency

Statistical feature selection methods like t-statistic filtering should be applied prior to graph model training. This reduces noise, improves training efficiency, and helps the model focus on meaningful financial attributes.

6. References

[1] S. T. King et al., "Credit Card Fraud Is a Computer Security Problem," IEEE Security & Privacy, vol. 19, no. 2, pp. 65-69, March-April 2021, doi: 10.1109/MSEC.2021.3050247.

[2] Delamaire, Linda, Hussein Abdou, and John Pointon. "Credit card fraud and detection techniques: a review." Banks and Bank systems 4.2 (2009): 57-68.

[3] M. A. Omar and D. Kiwanuka, "A State-of-the-Art Review of Machine Learning Techniques for Fraud Detection Research," in Proceedings of the 2018 International Conference on Computing and Big Data, Charleston, SC, USA, 2018, pp. 44-48.

[4] M. Isangediok and K. Gajamannage, "Fraud Detection Using Optimized Machine Learning Tools Under Imbalance Classes," arXiv preprint arXiv:2209.01642, 2022.

[5] P. Tiwari, S. Mehta, N. Sakhuja, J. Kumar, and A. K. Singh, "Credit Card Fraud Detection using Machine Learning: A Study," arXiv preprint arXiv:2108.10005, 2021.

[6] Hafez, I.Y., Hafez, A.Y., Saleh, A. et al. A systematic review of AI-enhanced techniques in credit card fraud detection. J Big Data 12, 6 (2025). https://doi.org/10.1186/s40537-024-01048-8.

[7] Parkar P, Bilimoria A. A survey on cyber security IDS using ML methods. 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 352–360, 2021. https://api.semanticscholar.org/CorpusID:235208042.

[8] V Bhusari, S Patil et al., “Application of Hidden Markov Model in Credit Card Fraud Detection,” International Journal of Distributed and Parallel Systems (IJDPS) Vol. 2, No. 6, November 2011.

[9] A. Srivastava et al., "Credit Card Fraud Detection Using Hidden Markov Model," IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 1, pp. 37–48, Jan. 2008.

[10] L.R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proc. IEEE, vol. 77, no. 2, pp. 257-286, 1989.

[11] S.S. Joshi and V.V. Phoha, “Investigate Hidden Markov Models Capabilities in Anomaly Detection,” Proc. 43rd ACM Ann. Southeast Regional Conf., vol. 1, pp. 98-103, 2005.

[12] Meng, W., Tischhauser, E.W., Wang, Q., Wang, Y. and Han, J. “When intrusion detection meets blockchain technology: a review,” in IEEE Access, March 2018.

[13] N. Alexopoulos, E. Vasilomanolakis, N.R. Ivanko, and M. Muhlhauser, “Towards blockchain-based collaborative intrusion detection systems,” in Proc. Int. Conf. Critical Inf. Infrastruct. Secur., 2017, pp. 1-12.

[14] J. Sotos and D. Houlding, “Blockchains for data sharing in clinical research: Trust in a trustless world,” Intel, Santa Clara, CA, USA, Blockchain Appl. Note 1, 2017.

[15] C.-T. Li, Y.-C. Tsai, C.-Y. Chen, and J. C. Liao, "Graph neural networks for tabular data learning: a survey with taxonomy and directions," arXiv, 2024, arXiv:2401.02143.

[16] S. Harish, C. Lakhanpal, and A. H. Jafari, "Leveraging graph-based learning for credit card fraud detection: a comparative study of classical, deep learning and graph-based approaches," Neural Computing and Applications, vol. 36, no. 34, pp. 21873-21883, 2024.

[17] D. Cheng, Y. Zou, S. Xiang, and C. Jiang, "Graph neural networks for financial fraud detection: a review," Frontiers of Computer Science, vol. 19, no. 9, pp. 1-15, 2025.

[18] Y. Yoo, D. Shin, D. Han, S. Kyeong, and J. Shin, "Medicare fraud detection using graph neural networks," in 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), July 2022, pp. 1-5

[19] J. Choi, J. Park, W. Kim, J. H. Park, Y. Suh, and M. Sung, "Pu GNN: Chargeback fraud detection in P2E MMORPGs via graph attention networks with imbalanced PU labels," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Cham: Springer Nature Switzerland, September 2023, pp. 243-258.

[20] L. Shammi, C. E. Shyni, S. Vinayagam, S. Aravindh, and J. S. Amalraj, "Fraud Detection in Accounting and Finance Enhanced by Knowledge-Driven GAT Networks," in 2024 First International Conference on Software, Systems and Information Technology (SSITCON), October 2024, pp. 1-5.

[21] T. Zhang and S. Gao, "Graph Attention Network Fraud Detection Based On Feature Aggregation," in 2022 4th International Conference on Intelligent Information Processing (IIP), October 2022, pp. 272-275.

[22] Fiore, U. et al. (2019). "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection." Information Sciences, 479, 448–455.

[23] Roy, A., & Sun, J. (2021). "Real-time Fraud Detection System Using Apache Flink and ML Models." IEEE Access.

[24] Sethi, T., Kantardzic, M. (2017). "Handling Concept Drift in Fraud Detection Using Active Learning and Dynamic Weighted Majority." International Journal of Data Science and Analytics.

[25] Wang, Y. (2025). "A Data Balancing and Ensemble Learning Approach for Credit Card Fraud Detection." arXiv preprint arXiv:2503.21160.

[26] B. Scholkopf and A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2001.

[27] S. Ghosh, D. Reilly, "Credit Card Fraud Detection with a Neural-Network," Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences, 1994, IEEE.

[28] P. Jiang, J. Zhang, and J. Zou, "Credit Card Fraud Detection Using Autoencoder Neural Network," in Proc. IEEE Int. Conf. Machine Learning and Applications, Western Ontario, Canada, 2025.

[29] S. Khatri, A. Arora, A. P. Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 680-683.

[30] L. Seyedhossein and M. R. Hashemi, "Mining information from credit card time series for timelier fraud detection," in 2010 5th International Symposium on Telecommunications (IST), Tehran, 2010, pp. 619–624.

[31] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, "Random Forest for Credit Card Fraud Detection," in 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 2018, pp. 1-6.

[32] I. D. Mienye and N. Jere, "Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions," IEEE Access, vol. 12, pp. 96893-96906, Jul. 2024.

[33] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, ''A neural network ensemble with feature engineering for improved credit card fraud detection,'' IEEE Access, vol. 10, pp. 16400–16407, 2022.

[34] I. D. Mienye and Y. Sun, ''A deep learning ensemble with data resampling for credit card fraud detection,'' IEEE Access, vol. 11, pp. 30628–30638, 2023.

[35] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi," Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 8, pp. 3784-3797, Aug. 2018.

[36] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

[37] Patidar, Raghavendra, and Lokesh Sharma. "Credit card fraud detection using neural networks." International Journal of Soft Computing and Engineering (IJSCE) 1.32-38 (2011).

[38] Mammen, Priyanka Mary. "Federated learning: Opportunities and challenges." arXiv preprint arXiv:2101.05428 (2021).

[39] Sazlı, M. H. (2006). A brief review of feed-forward neural networks. Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, 50(01). https://doi.org/10.1501/commua1-2_0000000026

[40] Salehinejad, Hojjat, et al. "Recent advances in recurrent neural networks." arXiv preprint arXiv:1801.01078 (2017).

[41] Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

[42] Devlin, Jacob. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[43] Morovat K, Panda B. A survey of artificial intelligence in cybersecurity. In 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 2020, pp. 109–115. https://doi.org/10.1109/CSCI51800.2020.00026.

[44] F. Shi and C. Zhao, "Enhancing financial fraud detection with hierarchical graph attention networks: A study on integrating local and extensive structural information," Finance Research Letters, vol. 58, p. 104458, 2023.