NATIONAL ECONOMICS UNIVERSITY
SCHOOL OF ADVANCED EDUCATION PROGRAMS

BACHELOR THESIS

Major: Business Administration

A Hybrid Approach To Anomaly Detection In Online Banking: Synthetic Behavioral Data And Comparative Model Analysis

Nguyen An Quynh

Ha Noi 2025

NATIONAL ECONOMICS UNIVERSITY
SCHOOL OF ADVANCED EDUCATION PROGRAMS

BACHELOR THESIS

A Hybrid Approach To Anomaly Detection In Online Banking: Synthetic Behavioral Data And Comparative Model Analysis

Student	**:**	**Nguyen An Quynh**
Student's ID	**:**	**11215084**
Major	**:**	**Business Administration**
Program	**:**	**Business Analytics**
Class	**:**	**BA63**
Supervisor	**:**	**PhD.Tran Hung**

Ha Noi 2025

ACKNOWLEDGEMENTS

This undergraduate thesis in Business Analytics with the topic "A Hybrid Approach To Anomaly Detection In Online Banking: Synthetic Behavioral Data And Comparative Model Analysis" is the result of my continuous efforts and the invaluable support and encouragement from my professors, family, friends, and colleagues. On this occasion, I would like to express my deepest gratitude to all those who have supported and accompanied me throughout the process of completing this thesis.

First and foremost, I would like to extend my sincere thanks to Drc. Trần Hùng - Faculty of Data Science and Artificial Intelligence and Head of DATCOM Lab, School of Technology, NEU who directly supervised my thesis, provided essential materials and scientific information, and offered dedicated guidance and support, helping me acquire the knowledge and skills necessary to complete this research.

I am also sincerely thankful to Mr. Bùi Tiến Lập, Vice Director of Loc Phat Bank, for providing me with valuable insights and real-world analysis regarding the current status of online banking transactions in Vietnam. His contributions were crucial in strengthening the practical foundation of my research.

I wish to extend my deep gratitude to Dr. Trần Đức Minh, Faculty of Data Science and Artificial Intelligence, School of Technology, National Economics University, and Dr. Vũ Lê Quỳnh Giang, Posts and Telecommunications Institute of Technology. Both have offered insightful support in building the theoretical framework and designing the research model. Their academic input, aligned with my direct supervisor's guidance, significantly enriched the quality and rigor of this study.

Additionally, I would like to express my heartfelt thanks to my family, who have always stood by my side, offering constant encouragement and unconditional support throughout my studies and during the writing of this thesis.

Moreover, I sincerely appreciate the assistance and support from my colleagues and friends at National Economics University, who created favorable conditions, shared valuable experiences, and provided helpful information that greatly contributed to the completion of this thesis.

Finally, I would like to thank all individuals and organizations not mentioned here by name but who have offered their support, encouragement, and assistance throughout my academic journey.

TABLE OF CONTENT

LIST OF TABLES AND FIGURES

List of Tables

List of Figures

LIST OF ABBREVIATIONS

Abbreviation	Full Term
AI	Artificial Intelligence
AML	Anti-Money Laundering
ANN	Artificial Neural Network
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BPTT	Backpropagation Through Time
CNN	Convolutional Neural Network
CTGAN	Conditional Tabular Generative Adversarial Network
DL	Deep Learning
DBN	Deep Belief Network
EDA	Exploratory Data Analysis
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
IEEE	Institute of Electrical and Electronics Engineers
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
NLP	Natural Language Processing
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
PU	Perceived Usefulness
PEOU	Perceived Ease of Use
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SAT	Satisfaction

SEM	Structural Equation Modeling
SERVQUAL	Service Quality (Responsiveness, Assurance, Reliability, Empathy, Tangibles)
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
TAM	Technology Acceptance Model
TimeGAN	Time-series Generative Adversarial Network
TVAE	Tabular Variational Autoencoder
VAE	Variational Autoencoder
XAI	Explainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting

EXECUTIVE SUMMARY

In the modern era of digital transformation, online banking has become an indispensable component of the global financial infrastructure. The rise of mobile applications, fintech platforms, and cloud-based services has fundamentally altered how individuals and institutions interact with financial systems. With these developments, however, has come a marked increase in the risk and prevalence of fraudulent activities. From identity theft and unauthorized transfers to more subtle patterns of financial manipulation and money laundering, online banking systems are now exposed to a wide spectrum of cyber and behavioral threats.

Financial transactions today are often high-frequency, irregularly timed, and multi-dimensional. They may reflect behavioral characteristics influenced by time-of-day, seasonal patterns, merchant categories, and customer history. Detecting fraud within this complex behavioral context requires not only identifying isolated outliers but also recognizing evolving patterns over time. However, financial institutions face several limitations in advancing fraud detection systems. First, there is a scarcity of publicly available and labeled transaction data due to privacy concerns. Second, anomalies such as fraud cases are extremely rare, leading to highly imbalanced datasets. Lastly, many models that perform well in academic contexts may fail in production due to poor interpretability and lack of robustness.

This study was undertaken in response to the growing need for robust, adaptable, and intelligent systems to detect anomalies in financial transactions. These anomalies may signal fraud, unauthorized access, or unusual behavioral patterns that deviate from expected norms. Traditional rule-based systems, while interpretable and easy to implement, suffer from rigidity and limited adaptability. Meanwhile, modern machine learning (ML) and deep learning (DL) techniques, despite their predictive power, often demand large labeled datasets and lack transparency. This research bridges these domains by introducing a hybrid model for anomaly detection that combines the strengths of temporal DL with interpretable ML classifiers, trained on synthetic data that mimics real-world behaviors in online banking. The thesis recognizes the limitations of traditional rule-based systems, which, although interpretable and domain-aligned, often suffer from rigidity and low adaptability. In contrast, ML and DL approaches offer scalability and flexibility but often at the expense of transparency and

reliability in data-sparse conditions. This study addresses these tensions by proposing a comprehensive three-phase framework that includes: (1) the generation of synthetic banking data via rule-based expert systems, (2) the application and evaluation of various detection algorithms includes machine learning, Long Short-Term Memory (LSTM)-based DL, and a hybrid model, and (3) a comparative performance analysis to determine the most effective approach under imbalanced data conditions.

A major innovation of this study is the creation of a synthetic dataset designed through a rule-based generation engine. The engine mimics real customer behavior, including transaction volumes, amounts, time patterns, and risk characteristics. Four categories of anomalies were encoded into the dataset based on industry practices: excessive monthly outflows, repeated short-interval transactions, structured money laundering schemes, and unauthorized account activities. A total of 1.5 million transactions were generated, encompassing thousands of synthetic users across daily and monthly timeframes. Another finding of this research underscores the superiority of hybrid models in fraud detection tasks involving behavioral sequences. While traditional ML is effective in capturing high-level patterns, it fails to model the temporal dynamics inherent in user behavior. Conversely, DL models like LSTM can capture complex sequence information but lack the interpretability and reliability needed for production systems. The proposed hybrid approach bridges this divide and leverages the strengths of both worlds.

Importantly, this study also demonstrates the practical value of synthetic data generation. By encoding expert-driven rules into synthetic behaviors, it becomes possible to generate large-scale, diverse, and annotated datasets that are suitable for supervised training. This alleviates the dependency on real labeled data and offers a framework for continuous model testing and validation.

INTRODUCTION

1. Research Rationale

The global shift toward digital banking has fundamentally transformed the financial services industry over the past two decades. Driven by advances in mobile technology, cloud computing, and fintech innovation, online banking has become the dominant channel for customer interaction. The history of online banking started in early 1980s as banks experimented with remote banking technologies. One of the first “home banking” systems launched by United American Bank in 1981 using personal computers and modems. Thirteen years later, in 1994, Stanford Federal Credit Union became the first financial institution to offer internet banking to all of its customers (Singh & Malhotra, 2004).

The early 2000s marked a critical phase in the proliferation of digital banking. With the increasing accessibility of the internet, banks rapidly expanded their online services. The widespread adoption of broadband infrastructure, improvements in web security protocols, and growing consumer trust significantly contributed to the mass adoption of online banking (Pikkarainen et al., 2004). By the end of 2010, over 70% of U.S. bank customers regularly used online banking services (FDIC, 2011). This digital momentum accelerated further with the global adoption of smartphones. The integration of mobile payment systems and banking applications into smartphones enabled customers to perform financial transactions anywhere, at any time. Financial institutions responded by developing mobile applications that offered real-time account access, peer-to-peer (P2P) payments, and biometric authentication (Tetnowski, 2021). Fintech challengers such as Revolut, Monzo, and N26 disrupted traditional banking models by offering digital-first services with minimal fees and friction (Zachariadis & Ozcan, 2017). As a result, features such as instant money transfers, QR code payments, e-wallets, and fingerprint or facial recognition became standard in modern banking experiences. Between 2010 and 2020, the number of mobile banking users grew at a compound annual growth rate (CAGR) exceeding 15%, reaching 2.1 billion globally by 2020 (Tetnowski, 2021). The COVID-19 pandemic acted as a catalyst for digital transformation. During this period, 40% of consumers globally adopted new digital banking services, as reported by McKinsey & Company (2021). Legacy banks were compelled to fast-track investments in real-time payments, artificial intelligence-based fraud detection, and virtual assistants. The World Bank (2022) highlighted that digital financial services significantly expanded financial inclusion in developing countries during the pandemic. As of 2023, more than 2.5 billion people globally are engaged in some form of digital banking, and this number is projected to exceed 3.6 billion by 2026 (Statista, 2023).

Vietnam, in the latter period, also experienced the same trend and rapidly emerged as a leader in digital banking adoption. According to the State Bank of Vietnam (SBV, 2023), over 95 million digital transactions are conducted daily. With a smartphone penetration rate exceeding 76% (Statista, 2023), major domestic banks such as Techcombank and MB Bank report that 90 to 95% of customer transactions occur through digital platforms (Vietnam Banks Association, 2023). Meanwhile, new entrants like Cake (VPBank) and TNEX (Maritime Bank) have embraced branchless, digital-only banking models targeting Gen Z and unbanked populations. These developments are closely aligned with the Vietnamese government's National Digital Transformation Program to 2025, which promotes the inclusive digitization of financial services (Vietnam Prime Minister’s Office, 2020).

However, this exponential growth in digital banking has also given rise to serious concerns about the misuse of technological vulnerabilities in banking systems for malicious purposes, particularly in fraud and money laundering. According to the Association of Certified Fraud Examiners (ACFE, 2022), the financial services industry remains the most targeted sector for fraud globally, accounting for nearly 20% of reported cases. The United Nations Office on Drugs and Crime (UNODC, 2021) estimates that 2 to 5% of global GDP, up to $2 trillion annually, is laundered through formal and informal financial channels, with an increasing proportion occurring via digital platforms. As digital infrastructure evolves, so too does the sophistication of financial crime schemes. In Vietnam, the Ministry of Public Security reported over 220,000 online scam and fraud cases between January and October 2024. These incidents often involved impersonation, investment scams, and unauthorized transactions. Social engineering remains a predominant tactic, with fraudsters exploiting user trust in government agencies and banks. Once victims disclose login credentials or OTPs, criminals rapidly execute multi-channel transfers, frequently before financial institutions can respond. Structural issues such as underreporting, long complaint processes, and low legal awareness among consumers exacerbate the challenge. Vietnamese banks also face growing compliance pressure with regard to anti-money laundering (AML) regulations, particularly following increased scrutiny by the Financial Action Task Force (FATF). A study by The Asian Banker (2023) identified Vietnam, Indonesia, and the Philippines as the most exposed countries in Southeast Asia to social-engineering-based bank fraud. Despite investments in cybersecurity infrastructure and AML compliance systems, many Vietnamese banks still rely on rule-based fraud detection approaches, which struggle to detect adaptive or multi-step fraud patterns.

Online banking’s ubiquity has thus been accompanied by the parallel growth of fraud and money laundering schemes. Financial institutions are under mounting pressure to deploy advanced anomaly detection systems capable of real-time threat identification and mitigation. However, one of the core challenges in developing such systems is the lack of accessible transactional data. Due to strict privacy regulations and institutional confidentiality, real banking datasets are rarely shared for research or model training purposes (Firat et al., 2023). Jensen et al. (2023) also confirm that no real banking data is publicly available for AML research because of the sensitive nature of financial transactions. This scarcity of data significantly hampers academic progress in fraud detection research, prompting the need for alternative, privacy-preserving methodologies.

In response to this research gap, the current study focuses on generating a synthetic dataset that simulates real-world transaction behaviors, including both normal and anomalous patterns. This thesis adopts synthetic data generation as a foundation for designing and evaluating fraud detection models that are relevant to the Vietnamese context. The synthetic dataset will reflect typical transaction behavior as well as rule-based injection of known fraudulent activities. Where applicable, fraud rules will align with SBV reporting thresholds and common scam archetypes observed in Vietnam. This tailored approach ensures contextual validity and prepares the model for future adaptation to live banking environments under appropriate regulatory oversight.

Following the construction of the synthetic dataset, the research proceeds to develop and compare three classes of prediction models for anomaly detection (1) Traditional ML models; (2) DL models; (3) Hybrid models. Each model will be evaluated using metrics such as accuracy, precision, recall, and F1-score to assess their effectiveness across multiple fraud scenarios. The ultimate goal is to present a scalable, explainable, and privacy-conscious solution that can support digital fraud detection in Vietnam and similar emerging digital economies.

2. Research Objectives and Questions

2.1. Research Objectives

The overarching objective of this study is to develop and evaluate an effective anomaly detection framework for online banking transactions using synthetic data, with a particular focus on replicating realistic user behavior and fraud patterns in the context of the Vietnamese financial sector.

From this overarching aim, the study defines the following specific objectives:

First, to establish a comprehensive theoretical for understanding normal and anomalous user behavior in online banking, including transaction frequency, timing, volume, and behavioral anomalies such as smurfing, dormant account activity, and location inconsistencies.

Second, to design and simulate a labeled synthetic dataset that reflects real-world online banking activity and embedded fraudulent behaviors, based on a rule-based framework informed by domain knowledge and regulatory guidelines (e.g., transaction thresholds, red flag indicators from AML policies).

Third, to develop, implement, and compare predictive models for anomaly detection using three methodological approaches: (1) traditional ML, (2) DL, and (3) hybrid models that combine the strengths of both.

Fourth, to evaluate the performance of each model using metrics such as accuracy, precision, recall, and F1-score, and analyze their applicability in identifying fraudulent transactions in Vietnamese digital banking environments.

Fifth, to provide practical insights and strategic recommendations based on the experimental findings, aimed at supporting banks and regulators in strengthening fraud detection systems without compromising data privacy.

2.2. Research Questions

In order to fulfill the objectives above, the study is guided by the following research questions:

1. What characterizes normal and anomalous transaction behaviors in online banking, and how can these behaviors be effectively simulated in a synthetic dataset?

2. What rule-based strategies can be used to generate synthetic data that closely mimic real banking transactions, while allowing for the injection of realistic fraud scenarios?

3. Which ML, DL, or hybrid modeling techniques are most suitable for detecting anomalies in the generated synthetic transaction dataset?

4. How do traditional ML models compare with DL models in terms of their ability to detect different types of anomalous behavior?

5. To what extent can the performance of anomaly detection models built on synthetic data generalize to real-world financial fraud cases in the Vietnamese banking context?

6. How can these insights inform future fraud detection strategies and regulatory compliance in Vietnam?

3. Research Subject and Scope

3.1. Research subject

The study focuses on modeling both normal user behavior and anomalous behavior such as high-value transactions, smurfing, location mismatches, frequent small transfers, and activity from dormant accounts through 7 original features: transaction id, transaction date, transaction time, sender account, receiver account, merchant id, bank channel, within a simulated digital banking environment.

3.2. Research Scope

3.2.1. Temporal Scope

The simulated transaction data spans a two-year period, emulating real-world online banking activities of 50 users.

3.2.2. Spatial Scope

This research is conducted in the context of Vietnam’s retail banking sector, with particular attention to how fraud manifests in local transaction behavior. The synthetic dataset structure, fraud rules, and evaluation criteria are tailored to reflect the regulatory and operational standards of Vietnamese banks, including typical thresholds for suspicious transaction reporting, types of domestic payment channels, and common fraud typologies reported in Vietnam. While the modeling techniques may have broader applications, the empirical validation and use case implementation are geographically scoped to Vietnam, aligning with national financial digitization strategies and fraud prevention needs.

3.2.3. Content Scope

This study focuses on transaction-level anomaly detection within internal online banking systems of a specific bank. Its scope encompasses: (1) simulation of realistic financial behavior and fraudulent activity over a two-year period for 50 users; (2) design and generation of labeled synthetic data, following the format and schema used by a Vietnamese bank; (3) construction and comparison of predictive models, including: traditional ML models (Random Forest, Extreme Gradient Boosting (XGBoost), Light gradient-Boosting Machine (LightGBM)), DL models (LSTM), hybrid models combining sequential and feature-based learning; (4) model evaluation using standard performance metrics (accuracy, precision, recall, F1-score).

The study excludes areas such as loan default prediction, credit risk scoring, or corporate banking analytics. It does not incorporate non-transactional behavioral data (e.g., device biometrics, voice data), nor does it utilize any real customer information, ensuring compliance with ethical research standards and data protection regulations.

4. Research Methodology

4.1. Data Collection Methods

4.1.1. Secondary Data

Secondary data for this study were gathered through desk research, involving a comprehensive review of existing literature, industry reports, and government publications. Statistical data were extracted from publicly available sources, including: annual reports of the State Bank of Vietnam, market research publications from consulting firms, sectoral studies on financial fraud and digital banking, reports issued by regulatory bodies such as the FATF, and statistical releases from official government portals,...

Additionally, conceptual frameworks and technical references were synthesized from academic journal articles, international conference proceedings, prior theses and dissertations, and research projects conducted at various institutional levels. These sources were used to inform the rule-based fraud definitions, the modeling architecture, and the synthetic data generation strategy.

4.1.2. Primary Data

Primary data for this research were obtained from two sources: (1) qualitative expert interviews and (2) a synthetically generated dataset based on expert-defined criteria.

Semi-structured interviews were conducted with banking professionals and risk management experts from commercial banks operating in Vietnam. The objective of these interviews was to gain domain-specific insights into the characteristics of both legitimate and anomalous transactions. Interviewees were asked to elaborate on:

- Behavioral traits typical of normal banking activity,
- Common indicators of suspicious or fraudulent transactions,

- Operational thresholds for triggering transaction monitoring alerts,
- Challenges with existing rule-based fraud detection systems.

The qualitative data obtained were analyzed thematically and served as the empirical basis for designing the rules that governed anomaly injection into the synthetic dataset.

Informed by the findings from expert interviews, a labeled synthetic dataset was generated to simulate two years of online banking activity for 50 users. The structure and attributes of the dataset closely follow the data schema of Vietnamese banks.

Anomalous transactions were systematically injected using a rule-based framework derived from expert validation. Fraud scenarios included high-value transfers, smurfing (frequent small transactions), sudden activity from dormant accounts, location-based inconsistencies, and temporal abnormalities. The simulated dataset was designed to be statistically realistic while maintaining full compliance with privacy and ethical standards, as no real customer information was used.

Details of the data simulation and injection methods are described in Chapter 3 (Synthetic Data Design).

4.2 Data Analysis Methods

4.2.1. Analysis of Secondary Data

Secondary data is compiled, controlled, arranged, compared, and contrasted according to different research purposes of the topics aimed at clarifying issues related to trends (statistics, or gaps, documents). Key findings from secondary data served as foundational input for defining fraud typologies and setting parameter thresholds for synthetic transaction generation.

4.2.2. Analysis of Primary (Synthetic) Data

The synthetic transaction data were analyzed using a multi-stage ML pipeline consisting of data preprocessing, model training, evaluation, and comparison. The analytical procedures included: data preprocessing, model implementation and model evaluation. Standard classification metrics were used, including Accuracy, Precision, Recall, and F1-Score. Confusion matrices and ROC-AUC curves were generated for comparative performance analysis.

5. Research structure

In addition to the list of abbreviations, tables, figures, references, and appendices, this thesis is structured into five main chapters: Chapter 1 outlines the theoretical framework; Chapter 2 reviews the literature and presents the proposed research model; Chapter 3 details the research methodology; Chapter 4 presents the research findings; and Chapter 5 discusses the results and outlines key research implications.

CHAPTER 1: THEORETICAL FRAMEWORK

1.1. Theoretical Framework of Machine Learning

1.1.1. Definition

Machine Learning (ML) is a subfield of artificial intelligence (AI) concerned with developing algorithms that enable computers to improve their performance on a task by learning from data, rather than through explicit programming. This classic definition, attributed to AI pioneer Arthur Samuel (1959), which characterizes ML as "the field of study that gives computers the ability to learn without being explicitly programmed". In essence, instead of a human programmer pre-defining all decision rules, an ML system automatically infers patterns and rules from experience (data). From this initial definition, a more formal and widely cited definition is given by Mitchell (1997), which considered ML as "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." This definition highlights three key elements of ML: the task (T) the system is trying to perform (e.g., classify a transaction as fraudulent or not), the experience (E) it learns from (e.g., historical labeled transaction data), and the performance measure (P) used to evaluate improvement (e.g., accuracy or error rate). Learning, in this context, means achieving better performance with more data over time. It is an operational definition of learning that focuses on measurable improvement, aligning with Alan Turing's perspective of replacing the question "Can machines think?" with "Can machines do what we (as thinking entities) can do?" In other words, ML systems operate primarily via induction, learning general rules from specific instances. This process is rooted in statistical inference and optimization theory. Unlike traditional programming paradigms that necessitate rule specification by human coders, ML models use algorithms to discover patterns and relationships directly from the data. These relationships are then used to make predictions or decisions on new, unseen instances. Statistically, this is often conceptualized as learning a function that maps inputs (features) to outputs (labels or predictions), typically by minimizing an objective function such as prediction error (Murphy, 2012).

What distinguishes ML with other statistical models is described by Breiman (2001), who famously delineated the two cultures of modeling: the data modeling culture, which relies on predefined statistical assumptions (e.g., linearity, normality), and the algorithmic modeling culture, which employs flexible algorithms to learn from data with fewer assumptions. In ML, the emphasis lies on predictive accuracy rather than interpretability. This shift has enabled the application of ML techniques to increasingly complex and high-dimensional data across a wide array of fields, including finance, medicine, and cybersecurity. While traditional statistical modeling often struggles when the number of predictors is very large relative to the number of observations or when relationships are highly nonlinear, ML techniques, bolstered by increases in computational power, are designed to handle large-scale, high-dimensional data and can automatically discover nonlinear relationships through techniques like kernel methods or deep neural networks. This makes ML well-suited for modern applications such as image recognition or web click-through prediction, where the patterns are too complex to specify manually. On the other hand, ML models often require large amounts of training data to perform well, whereas statistical models can sometimes give good insights even from smaller datasets if the model assumptions hold. In practice, the lines between ML and statistical modeling are blurring, leading to the emerging field of “statistical learning” that integrates both perspectives (Hastie et al., 2009). Nonetheless, ML’s general philosophy is more empirical and data-centric: let the data speak and find a model that yields the best predictive performance, using computational algorithms to search through model space. This philosophy has proven extremely powerful in real-world problem-solving, prompting even traditional statisticians to adopt ML techniques for complex tasks.

1.1.2. Historical Development

The historical development of ML reflects a dynamic intersection of AI, statistical modeling, and computational advances. From its inception in the mid-20th century to the sophisticated DL models of today, ML has undergone significant theoretical and practical transformations that have enabled its adoption in sectors ranging from finance and healthcare to cybersecurity and industrial automation.

The roots of ML can be traced to the work of Turing in the 1950s, who proposed the idea of "learning machines" capable of simulating aspects of human intelligence (Turing, 1950). Around the same time, Samuel (1959) demonstrated one of the first ML programs, a checkers-playing system that improved over time through a form of reinforcement learning. Samuel’s work marked a departure from rule-based systems and inspired a generation of research into pattern recognition and adaptive systems.

The 1960s and 1970s witnessed a rise in interest in symbolic AI and expert systems, which focused on encoding human expertise into rule-based programs. However, symbolic approaches struggled with scalability and adaptability. In contrast, ML researchers began to explore statistical models and neural networks. Rosenblatt (1958) development of the perceptron was an early step toward learning algorithms inspired by brain architecture. Despite initial optimism, the limitations of single-layer perceptrons particularly their inability to model non-linearly separable functions like XOR were highlighted by Minsky and Papert (1969), leading to a temporary decline in neural network research.

A resurgence of interest in the 1980s brought about several foundational advances. The backpropagation algorithm, popularized by Rumelhart et al. (1986), enabled the training of multi-layer neural networks and revived interest in connectionist models. Concurrently, researchers such as Quinlan (1986) developed decision tree algorithms, which offered interpretable models that could be applied to real-world datasets. These developments laid the groundwork for both theoretical understanding and practical applications.

The 1990s marked a significant shift towards statistical learning. Vapnik and Cortes (1995) introduced support vector machines (SVMs), which offered robust theoretical guarantees and performed well on high-dimensional data. Ensemble methods, such as bagging (Breiman, 1996) and boosting (Freund & Schapire, 1997), further improved prediction accuracy by combining the outputs of multiple models. During this period, the rise of Bayesian networks and probabilistic graphical models allowed for structured representation and inference under uncertainty (Pearl, 1988).

The 2000s ushered in the era of big data and scalable ML. The proliferation of data, coupled with improvements in computational power, led to the widespread adoption of ML in industry. The amount of data generated by the internet, sensors, and digital services grew exponentially, and ML algorithms were adapted to handle large-scale datasets. Support vector machines and neural networks were scaled up using techniques like kernel approximation and GPU computing, respectively. New algorithms suitable for big data emerged, such as gradient boosting machines (Friedman’s Gradient Boosting Trees in 2001 built on the boosting idea to allow sequential tree growth optimizing a differentiable loss). Variants like XGBoost (Chen & Guestrin, 2016) and later LightGBM

(Microsoft, 2017) further improved the efficiency of boosting, allowing models with hundreds of trees to be trained on millions of examples quickly. Ensemble methods (random forests, boosted trees) became go-to solutions for structured data problems, winning many ML competitions in the 2000s and 2010s due to their accuracy and robustness. Meanwhile, researchers continued to refine neural networks. Notably, in 2006, Hinton et al. introduced deep belief networks and unsupervised layer-wise pretraining, rekindling interest in deep neural architectures. The term "DL" began gaining traction for neural networks with many layers. A watershed moment came in 2012: a deep convolutional neural network (CNN) (Krizhevsky et al., 2012) dramatically won the ImageNet Large-Scale Visual Recognition Challenge by reducing the error rate by nearly half relative to the previous state-of-the-art (Snezana, 2025). After 2012, DL rapidly came to dominate benchmarks in computer vision, speech recognition, and later natural language processing (with recurrent neural networks and transformers). Tech companies invested heavily in DL for applications like image search, voice assistants, and translation.

Recent years have also seen the rise of explainable AI (XAI) techniques, addressing the need for transparency in high-stakes domains like finance. Tools such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) provide insights into model predictions, aiding both regulatory compliance and user trust. Additionally, federated learning and privacy-preserving techniques are emerging to allow collaborative model training across institutions without sharing sensitive data (Kairouz et al., 2021).

1.1.3. Algorithm Classification

In the contemporary landscape of computer science and data analytics, ML stands as a pivotal component of AI. One of the most foundational aspects in understanding ML lies in its classification into different learning paradigms, each determined by the nature of data used, the learning process, and the intended outcomes. The majority of academic literature and professional practice converges on a four-fold typology of ML algorithms: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (Goodfellow, Bengio, & Courville, 2016; Alpaydin, 2020; Russell & Norvig, 2021). These paradigms form the conceptual core of ML and serve as a framework for selecting and developing appropriate algorithms for real-world applications, including but not limited to anomaly detection, natural language processing, image recognition, and online financial services.

1.1.3.1. Supervised Learning

Supervised learning represents the most intuitive and widely used form of ML. In this paradigm, an algorithm is trained on a labeled dataset, meaning that each training example is paired with an output label. The model learns a mapping function from inputs (features) to outputs (target labels), allowing it to predict the output for unseen data (Mohri, Rostamizadeh, & Talwalkar, 2018).

The foundation for this approach can be traced back to regression analysis and discriminant analysis techniques of the 1960s and 1970s (Hastie, Tibshirani, & Friedman, 2009). These early methods laid the groundwork for more sophisticated algorithms such as decision trees, support vector machines (SVM), k-nearest neighbors (KNN), and later, ensemble methods like Random Forests and boosting algorithms (e.g., XGBoost, LightGBM).

A hallmark strength of supervised learning is its interpretability and validation framework. Model performance can be objectively evaluated using metrics such as accuracy, precision, recall, and area under the curve (AUC), which makes these models attractive in regulated environments. Additionally, the popularity of this approach also lies in its interpretability and robustness across domains, for example, classify skin cancer images with deep CNNs (Esteva et al.,2017) and predict disease progression and patient outcomes (Rajkomar et al., 2019) in medical diagnosis or the application of XGBoost and LightGBM in fraud detection (Liu et al., 2023).

1.1.3.2. Unsupervised Learning

Unsupervised learning provides an alternative to the supervised framework by operating on data without predefined labels. Unlike supervised learning, unsupervised learning deals with data that has no associated labels, which emerged to address the absence of labeled data in many real-world scenarios. This type of ML seeks to identify patterns or structures hidden within the data. Clustering and dimensionality reduction are the primary tasks under this category (Hastie et al., 2009). Historically, early clustering algorithms such as k-means and hierarchical clustering gained prominence in the 1980s, particularly in marketing and social science research. Over time, advancements in density-based methods like DBSCAN (Ester et al., 1996) and dimensionality reduction techniques such as Principal Component Analysis (PCA) and t-SNE expanded the frontier of unsupervised analytics. Modern unsupervised learning has expanded into DL with the advent of autoencoders and generative models. Variational autoencoders (Kingma & Welling, 2013) and generative adversarial networks (GANs) (Goodfellow et al., 2014) are prominent in image synthesis, data augmentation, and representation learning. These models learn latent representations that capture the underlying structure of the data, often serving as feature extractors for downstream tasks.

One critical application of unsupervised learning is anomaly detection, where models identify data points that deviate from normal behavior. The Isolation Forest algorithm, introduced by Liu et al. (2008), exemplifies an unsupervised technique that isolates anomalies through recursive partitioning. Another application of unsupervised learning can be found in genomics where unsupervised learning enables the exploration of high-dimensional gene expression datasets to detect unknown biomarkers (Ringnér, 2008) or clustering models classify emerging threat patterns based on behavioral signatures in cybersecurity (Liu, Ting, & Zhou, 2008), segmenting customers based on their purchasing behaviors in e-commerce (Yan et al., 2022).

A critical challenge in unsupervised learning lies in model evaluation, as the absence of ground truth complicates validation. Research has addressed this through the development of intrinsic metrics such as silhouette score and extrinsic evaluation using proxy tasks. Moreover, unsupervised models are being integrated into hybrid pipelines, such as pre-training language models before supervised fine-tuning or detecting anomalies before classification.

Recent advances in self-supervised learning blur the line between supervised and unsupervised paradigms. Models like SimCLR (Chen et al., 2020) and BYOL (Grill et al., 2020) train on proxy tasks (e.g., predicting image rotations) to learn useful representations, which are then applied in supervised tasks with fewer labels. These innovations have expanded the applicability of unsupervised learning to scenarios traditionally dominated by supervised methods.

1.1.3.3. Semi-supervised learning

Semi-supervised learning (SSL) occupies a critical intermediary position between supervised and unsupervised learning paradigms. It emerges in situations where labeled data is scarce or costly to obtain, but a vast quantity of unlabeled data is readily available. The foundational motivation behind SSL is the realization that leveraging the intrinsic structure present in vast quantities of unlabeled data can substantially enhance model generalization, reducing the dependency on costly or labor-intensive labeling processes. Early theoretical frameworks, such as self-training and co-training, crystallized during the 1990s, were instrumental in shaping the evolution of SSL methods. Blum and Mitchell's (1998) seminal work on co-training, where two classifiers iteratively label unlabeled examples for each other using different feature views, marked a turning point in formally recognizing the utility of unlabeled data.

The trajectory of SSL research expanded further with the introduction of self-training strategies, where a single classifier retrains itself on confidently labeled unlabeled data, an idea initially proposed by Yarowsky (1995). Concurrently, graph-based approaches emerged, with Zhu et al. (2003) proposing graph-based SSL frameworks that conceptualize the dataset as a graph, propagating label information through weighted edges based on feature similarity. These innovations demonstrated that careful exploitation of data topology could bridge the gap between fully supervised and unsupervised learning, setting the stage for subsequent algorithmic advancements. The biomedical domain has particularly benefited from semi-supervised learning methodologies. Zhou et al. (2021) demonstrated the effectiveness of semi-supervised graph convolutional networks (GCNs) in predicting protein functions, leveraging both known annotations and network-derived similarities to outperform traditional supervised techniques. Likewise, Zhang et al. (2022) illustrated the applicability of graph-based SSL for banking fraud detection, where transaction networks were utilized to identify fraudulent activities that conventional supervised methods failed to capture due to the rarity of labeled fraud instances.

The advent of DL further transformed the landscape of semi-supervised learning. Lee (2013) proposed pseudo-labeling, a simple yet impactful method where models assign "pseudo-labels" to unlabeled data based on their confident predictions, using these augmented labels during retraining. Building upon this foundation, Sohn et al. (2020) introduced FixMatch, which synergized pseudo-labeling with strong data augmentation and consistency regularization, achieving near-supervised performance on benchmarks like CIFAR-10.

Across domains, semi-supervised learning has realized transformative impacts. In healthcare, Cheplygina et al. (2019) reviewed how SSL techniques have enabled robust predictive modeling using incomplete electronic health records, particularly where privacy concerns limit full annotation. When looking at the field of autonomous driving, SSL frameworks like FixMatch have been instrumental in improving scene understanding from partially labeled street datasets (Sohn et al., 2020). Parallelly, SSL also shown it advance in natural language processing when Devlin et al. (2019) revolutionized the field with BERT, pre-training on massive unlabeled corpora via self-supervised objectives before fine-tuning with minimal labeled data, setting new state-of-the-art benchmarks across tasks.

1.1.3.4. Reinforcement Learning

Reinforcement Learning (RL) constitutes a distinct and powerful branch of ML wherein agents learn to make sequential decisions by interacting with an environment, receiving feedback through rewards or penalties. Unlike supervised learning, where an agent is provided with explicit labels indicating the correct output for each input, RL operates based on indirect, evaluative feedback from the environment. In RL, the agent interacts sequentially with an environment by observing states, selecting actions, and receiving rewards, which are scalar signals that assess the desirability of the resulting outcomes (Sutton & Barto, 2018). Rather than learning from fixed input-output pairs, the agent must infer, through experience, a policy that maximizes the cumulative expected rewards over time. Crucially, the feedback in reinforcement learning is often sparse, noisy, and delayed, posing additional challenges compared to supervised paradigms. For instance, in a game-playing scenario, an agent may make hundreds of moves without immediate feedback, only learning at the end whether its cumulative strategy led to a win or a loss. This delayed nature of reward necessitates the agent's ability to attribute credit to past actions, a phenomenon formalized through techniques like temporal-difference learning (Sutton, 1988). Consequently, reinforcement learning embodies a fundamentally different paradigm of feedback-driven learning, where the agent must simultaneously explore unknown actions and exploit accumulated knowledge to optimize long-term performance.

Foundational work by Sutton and Barto (1998) crystallized the formalization of RL problems through the framework of Markov Decision Processes (MDPs), establishing key concepts such as value functions, policy iteration, and temporal difference learning that continue to underpin modern advances. The early trajectory of RL research was then shaped significantly by methods like Q-learning (Watkins & Dayan, 1992), which introduced an off-policy approach capable of learning the optimal action-value function independently of the agent's current policy. Around the same period, policy gradient methods began gaining attention for their ability to optimize stochastic policies directly in continuous action spaces (Williams, 1992). These early methodologies laid the groundwork for the deep reinforcement learning renaissance that would emerge two decades later. With the advent of DL, reinforcement learning underwent a paradigm shift. Mnih et al.'s (2015) groundbreaking work on the Deep Q-Network (DQN) demonstrated that coupling Q-learning with deep CNNs allowed agents to master complex tasks such as playing Atari games at superhuman levels, using raw pixel inputs as the sole state representation. This milestone illustrated that deep RL could bridge the gap between low-level sensory perception and high-level decision-making, propelling a surge of interest and research investment into the field.

Applications of reinforcement learning have proliferated across a wide array of domains. One of the most crucial applications of RL can be found in robotics where it has been instrumental in teaching agents to perform dexterous manipulation tasks that are difficult to model explicitly (Levine et al., 2016). For instance, guided policy search has enabled robotic arms to autonomously learn intricate behaviors such as stacking blocks or threading needles, tasks traditionally requiring handcrafted control programs. In finance, Moody and Saffell (2001) applied reinforcement learning frameworks to optimize trading strategies, highlighting RL's potential in dynamic, stochastic environments where optimal decisions evolve over time. Healthcare has also witnessed transformative RL applications. Komorowski et al. (2018) demonstrated how RL models could recommend personalized treatment strategies for sepsis patients in intensive care units, learning policies from historical patient trajectories that outperformed clinician benchmarks in simulated evaluations. This work exemplifies the growing recognition that RL's ability to learn adaptive, sequential strategies is crucial for complex, high-stakes domains where static decision rules are insufficient. In the realm of natural language processing, reinforcement learning has underpinned advancements in dialogue systems and machine translation. Notably, Bahdanau et al. (2017) utilized actor-critic methods to improve neural machine translation models by directly optimizing sequence-level rewards, such as BLEU scores, rather than token-level cross-entropy losses. Similarly, Li et al. (2016) employed RL to enhance conversational agents, shaping dialogue generation toward more coherent and engaging interactions by modeling conversational goals explicitly through reward signals.

Recent years have also seen the rise of multi-agent reinforcement learning (MARL), where multiple agents learn and interact within shared environments. Lowe et al. (2017) introduced Multi-Agent Deep Deterministic Policy Gradient

(MADDPG), a framework that allows agents to learn decentralized policies while leveraging centralized critics during training. MARL finds critical applications in areas such as autonomous vehicle coordination (Shalev-Shwartz et al., 2016), resource allocation in wireless networks (Zhang et al., 2019), and cooperative robotic teams.

Despite these advances, reinforcement learning faces persistent challenges. Sample inefficiency remains a significant bottleneck; agents often require vast numbers of interactions to learn effective policies, rendering RL impractical for real-world applications with limited access to exploration opportunities (Dulac-Arnold et al., 2021). To address this, methods such as model-based RL (Ha & Schmidhuber, 2018) have been proposed, wherein agents first learn a model of the environment's dynamics and then plan within this model, drastically reducing the number of required real-world interactions. Another critical concern is stability and convergence. Training deep RL agents is notoriously brittle due to issues such as non-stationary targets, function approximation errors, and unstable feedback loops between policy updates and environment exploration (Henderson et al., 2018). Algorithms like Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Soft Actor-Critic (SAC) (Haarnoja et al., 2018) have emerged to enhance training stability, leveraging clipped objectives or entropy regularization to balance exploration and exploitation dynamically.

1.1.3.5. Comparative Insights and Application Considerations

The choice of classification algorithm depends on the specific requirements of the application. In highly regulated environments, interpretability may outweigh marginal improvements in predictive accuracy, making logistic regression and decision trees preferable. For tasks demanding high accuracy and resilience to noisy data, ensemble methods like random forests and gradient boosting are often more suitable.

Real-time fraud detection systems require not only high accuracy but also low latency and the ability to adapt to evolving fraud patterns. Ensemble methods have become particularly prominent in these contexts due to their balance of performance and flexibility. However, model selection should also consider the availability of labeled data, computational constraints, and the necessity for explanation.

In sum, the diversity of ML algorithms for classification allows practitioners to tailor models to the specific constraints and goals of fraud detection systems. The following section will explore how these algorithms are applied in practical anomaly detection settings across various domains.

1.2. Theoretical Framework of Deep Learning

1.2.1. Definition

The term "deep learning" itself gained academic prominence in the early 2000s, particularly with the work of Hinton et al. (2006), who described deep belief networks as a powerful method of learning multiple levels of representation. However, the foundational idea had already existed implicitly in multilayer perceptrons since the 1980s, and its broader conceptual roots extend back to biological and cognitive models of information processing.

DL is now formally defined as a subset of ML that employs artificial neural networks (ANNs) with multiple hidden layers to model complex patterns in data. These architectures automatically learn hierarchical features, transforming raw inputs into increasingly abstract representations that enhance decision-making capabilities (LeCun et al., 2015). Unlike traditional ML techniques that rely heavily on feature engineering, DL emphasizes end-to-end learning and adaptability to unstructured data. Psychologically, the appeal of DL can also be understood in terms of its alignment with human perception: just as humans recognize patterns through multiple stages: edges, shapes, objects, deep networks do so through stacked layers. This hierarchical nature is what differentiates DL from earlier, shallower models.

The definition of DL has also shifted and expanded with the rise of new architectures and application domains. While early definitions emphasized neural depth and supervised learning, modern understandings include unsupervised, self-supervised, and reinforcement learning strategies, as well as models such as CNNs, recurrent neural networks (RNNs), attention mechanisms, and transformers. Each architectural innovation has redefined what it means to learn deeply from data, pushing the boundary from perception tasks to sequential reasoning, and now, multi-modal synthesis.

1.2.2. Historical Development

The historical development of DL is a testament to the iterative and interdisciplinary nature of scientific progress. What began as theoretical neuroscience in the mid-20th century has transformed into one of the most influential technological revolutions of the 21st century. The field’s trajectory, from simple perceptrons to transformer-based architectures, demonstrates a continuous evolution shaped by new data availability, computational advances, and pressing societal needs. As DL continues to mature, its integration with fields such as econometrics, epidemiology, and systems biology will open new frontiers in predictive analytics and decision-making.

The inception of DL can be traced back to early models of neural computation. Culloch and Pitts (1943) introduced a simplified model of the neuron, proposing that any computable function could be represented by a network of these units. Their binary threshold logic model laid the theoretical groundwork for neural networks by showing how simple units could be combined to produce complex behavior. Following this research, Hebb (1949) proposed a learning rule for synaptic plasticity, suggesting that the strength of connections between neurons increased when they were activated simultaneously, a principle that inspired unsupervised learning algorithms. Rosenblatt (1958) operationalized these ideas through the perceptron, a single-layer neural network that could classify data based on linear decision boundaries. The perceptron generated considerable excitement and investment in the field of AI. However, the enthusiasm waned after Minsky and Papert (1969) influential critique in "Perceptrons," which demonstrated the limitations of single-layer models, particularly their inability to solve non-linearly separable problems such as the XOR function. This publication significantly curtailed research funding and ushered in what became known as the first "AI winter."

Interest in neural networks was reignited in the 1980s with the rediscovery and popularization of the backpropagation algorithm, which allowed for the training of multi-layer perceptrons (MLPs) through gradient descent (Rumelhart, Hinton, & Williams, 1986). This advancement marked a pivotal moment in DL history, as it enabled the training of deep architectures capable of capturing complex patterns in data. During this period, research began to diverge into specialized domains, including time-series forecasting, speech recognition, and medical diagnosis. Although computational limitations and lack of large-scale labeled datasets constrained practical implementation, the foundational work laid in this era-such as the introduction of CNNs by LeCun et al. (1989) for image recognition, would later serve as the basis for DL's resurgence.

The 2000s witnessed the convergence of three critical enablers: the rise of big data, the advent of general-purpose graphics processing units (GPUs) for training deep models, and algorithmic innovations that enhanced model stability and training efficiency. Hinton et al. (2006) introduced deep belief networks (DBNs), showcasing unsupervised pre-training as a strategy to overcome the challenges of vanishing gradients. The definitive breakthrough came in 2012, when Krizhevsky, Sutskever, and Hinton won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) using a deep CNN known as AlexNet (Krizhevsky et al., 2012). The model dramatically outperformed traditional computer vision approaches, igniting global interest in DL. This milestone catalyzed widespread investment in research and commercial applications. Simultaneously, research directions diversified. In finance, deep neural networks began to be explored for risk modeling and credit scoring (Sirignano, 2016). In medicine, CNNs demonstrated impressive accuracy in diagnosing diseases from medical images (Esteva et al., 2017). In cybersecurity, recurrent models began to track behavioral anomalies in network traffic (Kim et al., 2016).

With growing interest in modeling sequential data, especially in natural language processing and time-series forecasting, researchers revisited and refined recurrent neural networks (RNNs). Hochreiter and Schmidhuber (1997) research about LSTM model gained popularity for its ability to learn long-range dependencies, a challenge for traditional RNNs due to gradient vanishing. LSTM networks were adopted across multiple industries. They were used for patient monitoring using electronic health records in health care (Choi et al., 2016), to enable anomaly detection in transactional data by learning temporal spending behaviors (Jurgovsky et al., 2018) or to forecast traffic and optimize logistics,... The introduction of attention mechanisms by Bahdanau et al. (2014), and later the transformer architecture by Vaswani et al. (2017), marked a paradigm shift in DL for sequential data. Attention mechanisms allowed models to selectively focus on different parts of the input, improving interpretability and performance. Transformers soon replaced RNNs in many natural language tasks and began to penetrate other domains. For instance, financial researchers employed attention-based models to predict stock price movements using news headlines and temporal transaction patterns (Zhang et al., 2021).

Recent years have seen the proliferation of large-scale, pre-trained models such as BERT (Devlin et al., 2019) and GPT (Brown et al., 2020), which have further advanced the frontiers of DL. Transfer learning and self-supervised learning have become dominant themes, enabling models to be fine-tuned for specific tasks with minimal labeled data.

1.2.3. Application of Deep Learning in Time-Series Modeling

Time-series modeling represents one of the most crucial applications of DL, particularly due to the growing availability of high-frequency, high-dimensional sequential data in domains such as finance, healthcare, transportation, and industrial monitoring. DL offers a compelling advantage over traditional time-series techniques by its ability to model complex, non-linear dependencies and interactions across time steps without manual specification of lagged variables or temporal transformations (Fawaz et al., 2019).

Traditional time-series forecasting methods, such as ARIMA (Box & Jenkins, 1970), exponential smoothing, and state-space models, have long been the foundation of sequential modeling. While statistically sound, these methods often assume linearity and stationarity, limiting their ability to capture dynamic, high-frequency patterns or context-dependent anomalies. ML models such as support vector regression or random forests improved upon these constraints but still required extensive feature engineering and could not easily learn temporal hierarchies.

DL models, particularly those incorporating recurrence and convolution, have alleviated many of these constraints. Recurrent Neural Networks (RNNs), LSTMs, and Gated Recurrent Units (GRUs) were the first generation of deep temporal models to gain prominence, particularly for tasks involving temporal classification, regression, and sequence generation. These models can learn dependencies across arbitrary time intervals, making them ideal for irregular or non-linear time-series structures (Hochreiter & Schmidhuber, 1997). Beyond recurrent models, CNNs have been adapted for time-series tasks by treating sequences as one-dimensional signals. CNNs offer the advantage of parallelism and local pattern detection, capturing short-term trends and periodicities in the data. In domains like wearable health monitoring, CNNs have successfully identified arrhythmias or other physiological anomalies in ECG signals (Rajpurkar et al., 2017). Another powerful advantage of DL in time-series modeling lies in its ability to handle multivariate and multi-modal data while traditional time-series techniques struggle with high-dimensional inputs, particularly when variables interact in complex, non-linear ways. Deep architectures can model such interdependencies through joint embedding spaces and hierarchical abstraction.

Despite their advantages, deep time-series models are not without limitations. They require large amounts of data for effective training and are sensitive to distribution shifts, an important consideration in volatile environments like financial markets. Moreover, interpretability remains a challenge, particularly when decisions affect human welfare or financial outcomes.

1.2.4. Thrive into Recurrent Neural Networks (RNNs) and Long Short-Term Memory

Recurrent Neural Networks (RNNs) and their sophisticated variants, most notably Long Short-Term Memory (LSTM) networks, which represent cornerstone architectures in DL, specifically tailored for sequential data analysis. These models are designed to capture dependencies across time, making them particularly suited for time-series prediction, language modeling, and event sequence classification. Their theoretical origins, architectural innovations, and widespread applications mark a significant milestone in the development of DL as a tool for temporal intelligence.

RNNs are a class of neural networks where connections between units form a directed cycle, allowing the model to maintain an internal state or "memory" of previous inputs. This recursive property enables the modeling of dynamic temporal behavior where the prediction at time t depends on the inputs received at previous time steps (Elman, 1990). Mathematically, RNNs operate by updating their hidden state h_t at each time step based on the current input x_t and the previous state h_{t-1}, often using a non-linear activation such as tanh or ReLU. Despite their theoretical elegance, traditional RNNs face practical limitations, primarily due to the vanishing and exploding gradient problems during backpropagation through time (BPTT). These issues hinder their ability to learn long-term dependencies, particularly when sequences extend over many time steps (Bengio et al. 1994).

To overcome these challenges, Hochreiter and Schmidhuber (1997) proposed the LSTM architecture. Unlike conventional RNNs, LSTMs introduce memory cells and gating mechanisms, namely the input, output, and forget gates that regulate the flow of information. This structure allows the model to selectively retain or discard information, thereby preserving long-term dependencies over extended sequences. LSTMs have demonstrated empirical success across a multitude of domains, including language modeling, speech recognition, finance (Fischer & Krauss, 2018). Cho et al. (2014) introduced an alternative to LSTM called the Gated Recurrent Unit (GRU). GRUs simplify the LSTM architecture by combining the forget and input gates into a single update gate, and merging the cell state and hidden state. This results in fewer parameters and faster training, often without a significant loss in performance. GRUs have become popular in resource-constrained environments and have been successfully applied to short-term demand forecasting, real-time fraud detection, and conversational agents.

While RNNs and LSTMs are powerful, they are not without limitations. Their sequential nature limits parallelization during training, leading to longer training times compared to feedforward or attention-based models. Additionally, interpretability remains a concern, as understanding the internal state dynamics of recurrent models can be difficult. Recent research efforts have aimed at integrating attention mechanisms with LSTM architectures to enhance interpretability and performance. The attention-based LSTM framework allows the model to weigh the importance of past hidden states dynamically, thus providing insights into temporal relevance (Bahdanau, Cho, & Bengio, 2014). Moreover, techniques such as neural ordinary differential equations (Neural ODEs) and memory-augmented neural networks offer promising directions for overcoming RNN limitations (Chen et al., 2018).

1.2.5. The combination of traditional Machine Learning and Deep Learning

The concept of hybrid systems has evolved in parallel with advances in AI and systems engineering. In the 1990s, early hybrid systems were developed in expert systems and control systems, where rule-based reasoning was combined with statistical inference to improve flexibility (Boussaïd et al., 2013). These early approaches laid the groundwork for more sophisticated integrations seen in modern ML.

The development of hybrid ML models gained traction in the 2000s, particularly with the growth of neural-symbolic systems that sought to combine the learning ability of neural networks with the logic structure of symbolic reasoning (Garcez et al., 2009). The widespread adoption of DL in the 2010s further propelled hybrid methods, as practitioners realized that deep neural networks, while powerful, often lacked interpretability and were data-intensive. To address this, researchers began combining the strengths of both worlds. Fiore et al. (2019) developed a hybrid model that used deep autoencoders to extract features from transactional sequences, which were then classified using support vector machines. Similarly in the field of credit card fraud detection, Chen et al. (2020) also proposed a framework where LSTM-derived embeddings were concatenated with domain-specific features and passed to gradient boosting classifiers, improving performance. More recently, hybrid architectures have incorporated attention mechanisms, transformers, and graph neural networks.

These models extend traditional hybridization by learning not only temporal and spatial features but also relational dependencies in financial transaction networks (Zhang et al., 2021). Such developments demonstrate the continual evolution of hybrid systems in response to increasing data complexity and operational requirements.

1.3. Theoretical Framework of Time-series Analysis

1.3.1. Definition

Time-series analytics refers to the set of statistical, mathematical, and computational methods used to analyze data points collected or recorded at specific time intervals. Unlike cross-sectional data, where observations are independent and identically distributed, time-series data exhibit temporal dependencies, trends, seasonality, and autocorrelation that must be accounted for in modeling and inference (Hamilton, 1994). The core objective of time-series analytics is to extract meaningful characteristics from temporal data to understand past behavior and forecast future values. This is especially critical in dynamic environments such as finance, healthcare, manufacturing, and climate science, where real-time insights and predictive accuracy can yield substantial strategic benefits (Box, Jenkins, & Reinsel, 2015).

Time-series analytics can be broadly divided into descriptive analytics (summarizing data patterns over time), diagnostic analytics (explaining causes of temporal anomalies), predictive analytics (forecasting future states), and prescriptive analytics (recommending actions based on temporal forecasts). These categories have evolved over time, fueled by the convergence of traditional statistics with ML and AI.

1.3.2. Historical Development of Time-Series Analytics

The historical development of time-series analytics reflects an evolving understanding of how temporal data can be harnessed to uncover structure, predict change, and support decision-making. The earliest uses of time-series data can be traced back to astronomy and economics in the 17th and 18th centuries, where researchers manually recorded planetary movements and commodity prices to identify trends and cycles (Anderson, 1971).

The formalization of time-series analysis began in the early 20th century with the development of statistical techniques to quantify temporal patterns. A major milestone was Yule's (1927) development of autoregressive (AR) models to explain sunspot cycles. This laid the groundwork for stochastic process modeling, particularly the Autoregressive Moving Average (ARMA) family of models. Slutsky (1937) further expanded the field by introducing concepts of exogenous and moving average components in economic data. The next major advancement came with Box and Jenkins' (1970) systematization of Autoregressive Integrated Moving Average (ARIMA) modeling, which became the standard framework for time-series forecasting in fields such as economics, industrial process control, and meteorology. Their methodology emphasized model identification, parameter estimation, and diagnostic checking, a systematic cycle that remains foundational in classical time-series modeling today.

During the 1980s and 1990s, the field was influenced heavily by the introduction of state-space models and the Kalman filter, enabling real-time updating of forecasts. These tools were especially impactful in financial econometrics, allowing analysts to handle irregularly spaced or incomplete data more robustly (Harvey, 1990). Parallel developments in frequency domain analysis, particularly spectral analysis and Fourier transforms, allowed researchers to analyze time-series data through their cyclical components. These techniques became essential for understanding seasonality and cyclicity in economic indicators, weather data, and signal processing applications (Bloomfield, 2000).

The late 1990s and early 2000s saw the integration of ML into time-series analysis. Regression trees, support vector machines, and ensemble methods were introduced to handle nonlinearities and higher-dimensional temporal features. However, these methods still required extensive manual feature engineering and lacked mechanisms to learn temporal dynamics directly. The arrival of DL in the 2010s marked a turning point in time-series analytics. Recurrent Neural Networks (RNNs), and later LSTM models, allowed for end-to-end learning from raw sequences, eliminating the need for hand-crafted lag features. In parallel, attention mechanisms and transformers enabled models to capture long-range dependencies and contextual interactions more effectively than classical models (Vaswani et al., 2017).

1.3.3. Application of Machine Learning and Deep Learning in Time-Series Analytics

In recent years, the field of time-series analytics has witnessed a paradigmatic transformation through the integration of ML and DL techniques. This transition, reflected across a breadth of academic and applied research, signals a shift from traditional linear modeling frameworks to data-driven, high-capacity models capable of capturing complex temporal relationships in sequential data.

Supervised ML models, including decision trees, random forests, support vector machines (SVM), and gradient boosting methods, have become prevalent in time-series forecasting and classification tasks. These models are particularly effective when time-series data are converted into supervised learning formats through feature extraction (e.g., lagged values, rolling means, seasonal indicators). For example, random forests and boosting models have been widely used in credit risk modeling, energy demand forecasting, and customer churn prediction. They perform well on structured data, offer interpretability through feature importance, and scale to large datasets. However, they typically require careful feature engineering and may struggle with capturing temporal dependencies inherently present in sequential data (Bontempi et al., 2013).

Another dominant trajectory is the application of deep neural architectures, particularly recurrent neural networks (RNNs), LSTM networks (LSTMs), and gated recurrent units (GRUs), which enable the modeling of sequential dependencies directly from raw data. Across financial anomaly detection, patient health trajectory modeling, and traffic forecasting, these models consistently outperform traditional AR techniques when data exhibit long memory and temporal non-stationarity (Choi et al., 2016; Fischer & Krauss, 2018; Ma et al., 2015). A meta-analytic synthesis suggests that LSTMs offer the most robust results where input sequences are long, irregular, and characterized by variable lags between events. The emergence of attention-based models and transformers further advances this frontier, particularly for tasks involving multi-horizon forecasting or simultaneous prediction across correlated time-series. These architectures, originally pioneered in natural language processing, have been reconfigured to accommodate structured time-series inputs with varying granularity and frequency. Meta-studies indicate that transformers offer higher forecasting accuracy and computational scalability in domains ranging from inventory management and power grid optimization to biomedical signal processing (Zhou et al., 2021; Roy et al., 2019; Li et al., 2021).

What is striking in the growing body of cross-domain evidence is the convergence toward hybrid and ensemble approaches. These systems combine feature extraction and classification layers from different paradigms, for example, using LSTM-based embeddings fed into tree-based classifiers or stacking CNNs with gradient boosting methods. Such architectures are shown to yield consistent improvements in predictive accuracy, robustness to noise, and model interpretability. Case studies from smart cities, environmental monitoring, and industrial quality control validate the generalizability of these architectures across domains with varying data structures, temporal resolutions, and operational constraints (Zhang et al., 2022).

Despite these advances, current study also highlights several persistent challenges: (1) the scarcity of large, labeled time-series datasets in many domains; (2) the problem of distributional shifts and concept drift in streaming environments; and (3) the opacity of deep models in high-stakes decision-making contexts. While attention visualization and feature attribution techniques such as SHAP and Integrated Gradients are increasingly applied, their integration into real-time decision systems remains limited.

1.4. Theoretical Framework of Anomaly Detection

1.4.1. Definition

Anomaly detection, sometimes referred to as outlier detection, is a critical research area in data science, statistics, and AI. Although the terminology “anomaly detection” is relatively modern, the conceptual underpinnings of the field can be traced back to early statistical research in the 19th and 20th centuries. One of the earliest contributions came from Peirce (1852), who proposed a method to discard outliers from experimental data in physics. By the mid-20th century, robust statistical methods like Grubbs' Test (1950), Tukey’s box plot method (1977), and Mahalanobis distance (1936) provided systematic ways to identify extreme values within univariate and multivariate data (Barnett & Lewis, 1978). A pivotal moment in the conceptual framing of anomaly detection occurred with the work of Hawkins (1980), who defined an outlier as "an observation which deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". This interpretation not only formalized the idea of deviation but introduced the notion that anomalies could be symptomatic of underlying structural irregularities potentially emergent threats, mistakes, or opportunities (Hawkins, 1980). As computational capabilities expanded, this notion found relevance in multiple domains, particularly those dealing with high-volume or high-velocity data.

Since an anomaly can represent a critical piece of information. Depending on context, it may signal a bank fraud, a network intrusion, a malfunction in an engine, or a medical condition. Early work in statistics treated anomalies as noise to be removed to improve the fit of predictive models. However, in modern data analysis these outliers often are the primary interest, as they may correspond to important but infrequent events. The challenge lies in differentiating true anomalies from mere noise or benign deviations. From a functional perspective, anomaly detection can be formalized by defining a scoring function f(x) that assigns an anomaly score to each data point x. An anomaly is detected if the score exceeds a threshold τ (i.e., if f(x)>τ for some point x, then x is flagged as anomalous)

1.4.2. Historical Development of Time-Series Analytics

Anomaly detection, as a formal research area, has evolved significantly across disciplines such as statistics, computer science, and AI. This development spans over 2 decades, transitioning from classical statistical methods to advanced DL and hybrid systems. While the applications have become increasingly domain-specific, the financial sector has emerged as a major field of interest due to the immense consequences of undetected anomalies in high-stakes environments such as fraud prevention, AML, and risk management.

The historical roots of anomaly detection can be traced back to the mid-19th century. Charles Sanders Peirce (1852) introduced a method for eliminating observational errors, laying an initial foundation for the statistical treatment of outliers. The concept gained traction with the development of formal statistical tools designed to detect deviations within univariate and multivariate data. These tools include Mahalanobis distance (Mahalanobis, 1936), which measures the distance between a point and a distribution, Grubbs' test for univariate outlier detection (Grubbs, 1950), and Tukey's EDA that utilized boxplots to identify anomalies (Tukey, 1977). The seminal work by Hawkins (1980) marked a transformative milestone for the sector when impling that anomalies were not mere statistical noise but potential indicators of critical, latent phenomena. Such an understanding was pivotal, particularly in areas such as quality control and process optimization.

As computational power expanded in the late 20th century, anomaly detection methodologies diversified. Chandola et al. (2009) classified techniques into three categories: supervised, semi-supervised, and unsupervised learning. This categorization reflected the varying levels of labeled data available in different real-world scenarios. Supervised approaches frame anomaly detection as a binary classification problem using labeled data; semi-supervised methods learn from normal data and identify deviations; while unsupervised approaches identify anomalies based on deviation from dense clusters or learned reconstruction errors. The latter is particularly useful when labeled examples of anomalies are rare or non-existent. A landmark innovation was the Isolation Forest algorithm proposed by Liu et al. (2008), which isolates anomalies through recursive random partitioning. It became highly influential in high-dimensional financial and cybersecurity contexts.

With the advent of DL in the 2010s, researchers began to model non-linear, high-dimensional data distributions more effectively. Autoencoders and variational autoencoders (VAEs) became prominent tools for learning representations of normal data and flagging deviations based on reconstruction errors. Recurrent neural networks (RNNs), especially LSTM networks, gained popularity for sequential anomaly detection. Originally proposed by Hochreiter and Schmidhuber (1997), LSTMs proved useful in modeling long-term dependencies in temporal data. In financial applications, LSTMs have been used to detect abnormal transaction sequences and forecast account activities (Jurgovsky et al., 2018). Sequence-to-sequence models and attention-based architectures, such as Transformers (Vaswani et al., 2017), further advanced the capacity for multi-horizon anomaly detection. Their success in natural language processing (NLP) was soon translated to time-series domains, including stock price movement prediction and algorithmic trade monitoring.

Recently, across domains, there has been a convergence toward hybrid systems, integrating statistical models, rule-based logic, and ML. These systems combine the high accuracy of deep models with the transparency and domain knowledge embedded in rule systems. For instance, in financial anomaly detection, behavioral risk scores generated from LSTMs may be passed through expert-crafted thresholds to generate alerts. Academic and industrial research has also highlighted the importance of combining anomaly detection with graph analytics. Financial networks, such as those representing transfers between accounts, benefit from graph neural networks (GNNs) to detect structurally anomalous behavior

Despite significant progress, several challenges remain. Class imbalance continues to hinder the performance of standard classification models. Methods like cost-sensitive learning, focal loss, and synthetic oversampling are employed, but often with mixed results in dynamic environments (Bahnsen et al., 2016). Concept drift is another pressing concern that models trained on past data may become obsolete as behavior evolves. Online learning, periodic retraining, and adaptive model architectures are active areas of research aimed at addressing this issue. Moreover, privacy and fairness considerations are increasingly emphasized in anomaly detection pipelines, particularly in financial institutions subject to regulatory scrutiny.

1.4.2. Application of Machine Learning and Deep Learning in Time-Series Analytics

In recent years, the field of time-series analytics has witnessed a paradigmatic transformation through the integration of ML and DL techniques. This transition, reflected across a breadth of academic and applied research, signals a shift from traditional linear modeling frameworks to data-driven, high-capacity models capable of capturing complex temporal relationships in sequential data.

Supervised ML models, including decision trees, random forests, support vector machines (SVM), and gradient boosting methods, have become prevalent in time-series forecasting and classification tasks. These models are particularly effective when time-series data are converted into supervised learning formats through feature extraction (e.g., lagged values, rolling means, seasonal indicators). For example, random forests and boosting models have been widely used in credit risk modeling, energy demand forecasting, and customer churn prediction. They perform well on structured data, offer interpretability through feature importance, and scale to large datasets. However, they typically require careful feature engineering and may struggle with capturing temporal dependencies inherently present in sequential data (Bontempi et al., 2013).

Another dominant trajectory is the application of deep neural architectures, particularly recurrent neural networks (RNNs), LSTM networks (LSTMs), and gated recurrent units (GRUs), which enable the modeling of sequential dependencies directly from raw data. Across financial anomaly detection, patient health trajectory modeling, and traffic forecasting, these models consistently outperform traditional AR techniques when data exhibit long memory and temporal non-stationarity (Choi et al., 2016; Fischer & Krauss, 2018; Ma et al., 2015). A meta-analytic synthesis suggests that LSTMs offer the most robust results where input sequences are long, irregular, and characterized by variable lags between events. The emergence of attention-based models and transformers further advances this frontier, particularly for tasks involving multi-horizon forecasting or simultaneous prediction across correlated time-series. These architectures, originally pioneered in natural language processing, have been reconfigured to accommodate structured time-series inputs with varying granularity and frequency. Meta-studies indicate that transformers offer higher forecasting accuracy and computational scalability in domains ranging from inventory management and power grid optimization to biomedical signal processing (Zhou et al., 2021; Roy et al., 2019; Li et al., 2021).

What is striking in the growing body of cross-domain evidence is the convergence toward hybrid and ensemble approaches. These systems combine feature extraction and classification layers from different paradigms, for example, using LSTM-based embeddings fed into tree-based classifiers or stacking CNNs with gradient boosting methods. Such architectures are shown to yield consistent improvements in predictive accuracy, robustness to noise, and model interpretability. Case studies from smart cities, environmental monitoring, and industrial quality control validate the generalizability of these architectures across domains with varying data structures, temporal resolutions, and operational constraints (Zhang et al., 2022).

Despite these advances, current study also highlights several persistent challenges: (1) the scarcity of large, labeled time-series datasets in many domains; (2) the problem of distributional shifts and concept drift in streaming environments; and (3) the opacity of deep models in high-stakes decision-making contexts. While attention visualization and feature attribution techniques such as SHAP and Integrated Gradients are increasingly applied, their integration into real-time decision systems remains limited.

Conclusion, anomaly detection has become a critical capability in a wide range of industries, serving roles from fraud prevention and patient safety to infrastructure resilience and cybersecurity defense. While each domain has its specific challenges and requirements, the core methodologies, especially those leveraging DL and hybrid architectures, have proven to be highly adaptable. As data complexity continues to grow, anomaly detection systems must evolve to remain accurate, interpretable, and responsive.

CHAPTER 2: LITERATURE REVIEW

2.1. Literature Review

2.1.1. Literature Review for Generating Synthetic Data for Online Banking Transactions

2.1.1.1. Review of International Studies

The generation of synthetic data for financial applications, particularly in the domain of online banking transactions, has undergone significant evolution in the past three decades. This progression reflects a broader transformation in how data is leveraged for ML, privacy compliance, and model development in the digital finance era.

The earliest motivations for generating synthetic data in finance were rooted in concerns about data availability and privacy. With the advent of large-scale electronic financial systems in the 1990s, researchers and institutions began to recognize the limitations of relying solely on real-world datasets for model development and testing. Financial data was often proprietary, sensitive, or subject to strict regulatory constraints, making it inaccessible for academic or experimental purposes (Winkler, 1993). The early phase of synthetic data research in finance was dominated by resampling techniques and statistical simulations, particularly bootstrapping and Monte Carlo methods (Rubin, 1987; Efron & Tibshirani, 1993). While these approaches provided a framework for testing hypotheses under uncertainty, they lacked the structural complexity and behavioral realism required for transaction-level simulation.

With the widespread adoption of ML in fraud detection, credit scoring, and risk analytics in the early 2010s, the demand for large and representative datasets surged. Yet, access to real banking data remained constrained due to regulatory barriers such as the General Data Protection Regulation (GDPR) in the European Union and similar frameworks in North America and Asia, which led to the formalization of synthetic data generation as a subfield in AI and data privacy research. In the early stages, oversampling techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) became widely adopted to address class imbalance in fraud detection. While not full-fledged synthetic data generators, SMOTE and its variants provided an early bridge between statistical data augmentation and ML (Chawla et al., 2002). Although SMOTE does not generate fully synthetic records with contextual dependencies, it marked an important transition to algorithmically generated synthetic examples.

In the 2010s, Agent-Based Modeling (ABM) emerged as a prominent approach for generating synthetic financial transaction data, particularly in the context of fraud detection and financial crime analysis. Unlike traditional data augmentation techniques, ABM enables the creation of complete, realistic datasets from scratch, without requiring an existing source dataset, by simulating interactions among autonomous agents within defined financial ecosystems. The popular of ABM frameworks for financial transactions began with the development of BankSim in 2014, which provided a foundational framework for modeling typical and fraudulent behaviors in traditional banking environments (Lopez-Rojas & Axelsson, 2014). Expanding upon this approach, PaySim (2016) adapted ABM to the mobile money ecosystem, modeling transactional patterns in systems like M-Pesa. By incorporating hierarchical agent roles and mobile-specific transaction types, such as cash-in, cash-out, and peer-to-peer transfers which enabled more realistic simulation of user-driven financial behavior in developing economies (Lopez-Rojas & Axelsson, 2016). The focus then shifted toward credit card transactions with the introduction of CardSim (2020), which emphasized behavioral profiling and spending constraints, which focus on average transaction amount for each user, which deal with the specific characteristic of high-volume, low-value credit card ecosystems, where imbalanced datasets and sparse fraud labels hinder machine learning effectiveness. In contrast to these user-centric models, AMLSim (2018–2019) represented a paradigm shift by modeling financial institutions and accounts as nodes within a graph structure. Designed for anti-money laundering (AML) research, AMLSim simulated typologies like layering and structuring, enabling the generation of scalable, graph-based datasets annotated with Suspicious Activity Report (SAR) labels (Weber et al., 2018).

In the same period, with the advancement of DL, generative models began to dominate synthetic data research. The introduction of GANs by Goodfellow et al. (2014) revolutionized the ability to learn and replicate high-dimensional data distributions, which comprise a generator and a discriminator engaged in adversarial training, where the generator learns to produce indistinguishable data samples from real distributions. CTGAN (Xu et al., 2019), which modifies the GAN architecture to better handle tabular data, and TimeGAN (Yoon et al., 2019), which addresses sequential dependencies in time-series, extended the applicability of generative modeling to financial transactions. Otherwise, another popular method in this aspect called VAEs introduced by Kingma and Welling

(2013), which leverage an encoder-decoder architecture to model latent distributions and generate synthetic data with high fidelity. In recent year, Karst et al. (2024), in the effort to provide a comparative benchmark of CTGAN, TVAE, and DoppelGANger in generating synthetic financial transactions, had concluded that no single generative approach is universally superior across dimensions of utility, privacy, and structural fidelity, though each has advantages in specific contexts. Despite the outstanding of GANs, VAEs and other DL-based methods, these generative models require access to large volumes of representative real-world data, which paradoxically contradicts the very motivation behind their usage in data-scarce environments. Moreover, without explicit conditioning mechanisms, these models often struggle to capture rare but critical anomaly patterns.

2.1.1.2. Review of Domestic Studies

To the best of the author's knowledge, there appears to be a lack of publicly accessible Vietnamese academic research specifically focused on generating synthetic data for online banking transactions. While studies such as the one by Cao and Do (2014) have explored data mining techniques like the CLOPE algorithm for detecting money laundering in Vietnam's banking industry, they do not delve into the generation of synthetic data. Similarly, research on internet banking adoption in Vietnam, such as the study by Nguyen-Viet and Huynh (2021), focuses on consumer behavior rather than synthetic data generation .

2.1.2. Literature Review of Anomaly Detection for Online Banking Transactions.

2.1.2.1. Review of International Studies

The evolution of anomaly detection techniques in online banking has transitioned from rule-based systems to sophisticated DL and hybrid architectures, driven by the increasing digitalization of banking services, the diversity and complexity of fraud schemes, and the dynamic nature of user behavior (Achituve et al., 2019; Cui et al., 2021). Online banking differs fundamentally from traditional financial domains due to the prevalence of non-monetary events such as profile updates, device changes, and login activities, all of which play a crucial role in fraudulent activities but are not captured by models focusing solely on monetary transactions. Furthermore, the presence of extreme class imbalance, limited labeled fraud examples, and the demand for real-time detection render the problem particularly challenging and require more adaptive and intelligent detection mechanisms (Bhattacharyya et al., 2011).

Historically, traditional ML techniques including logistic regression, decision trees, support vector machines, and ensemble methods such as random forests and gradient boosting frameworks like XGBoost and LightGBM have formed the backbone of many fraud detection systems (Chen & Guestrin, 2016; Jurgovsky et al., 2018). These models are favored for their interpretability and computational efficiency. However, they treat each transaction independently, ignoring the temporal dynamics critical in online fraud detection. Recent innovations have sought to overcome this through DL techniques such as Recurrent Neural Networks (RNNs), particularly LSTM and Gated Recurrent Units (GRUs), which model sequences of user behavior and are more effective in capturing long-term dependencies and behavioral patterns (Jurgovsky et al., 2018; Li et al., 2018).

Transformer-based models, which have revolutionized natural language processing, have also begun to be adopted in financial fraud detection due to their ability to model long-range dependencies in data more efficiently than RNNs (Moreira et al., 2022). These models operate in parallel and can handle complex patterns across longer transaction sequences. In parallel, unsupervised and semi-supervised methods, such as autoencoders, variational autoencoders, and GANs, are increasingly employed to detect anomalies without relying heavily on labeled data (Xu et al., 2019; Wang et al., 2018). These models learn representations of normal behavior and flag deviations as potential anomalies. GANs, in particular, are used to generate synthetic fraudulent samples, improving training on imbalanced datasets.

Another powerful approach involves hybrid architectures that combine LSTM-based DL feature extractors with traditional ML classifiers like XGBoost or Random Forests (Li et al., 2018; Abbassi et al., 2024). These hybrids aim to integrate the strengths of both worlds: the temporal modeling capacity of LSTMs and the interpretability and efficiency of tree-based classifiers. Such combinations have shown significant improvements in detection performance and adaptability, especially in online banking contexts where data is heterogeneous and sparse.

Graph-based anomaly detection has also gained attention, particularly for detecting coordinated fraud involving multiple accounts. GNNs and GCNs are effective at capturing relational structures in transactional data, such as shared devices or repeated transfer paths among different accounts (Moreira et al., 2022). Given regulatory pressures and the operational need for explainability, there is a strong emphasis on model transparency. Methods such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are widely used to provide insights into model decisions. Furthermore, attention mechanisms in models like hierarchical LSTM enable interpretable sequence modeling by highlighting the most relevant parts of transaction histories contributing to a fraud decision (Achituve et al., 2019).

Deploying these models in real-world banking systems introduces practical challenges such as latency, data integration, and retraining. Many institutions address this by implementing two-stage systems combining a lightweight model for real-time screening with a complex model for deeper analysis. Modular hybrid systems also allow for component-wise updates, supporting agile responses to emerging fraud patterns.

Evaluation of fraud detection models increasingly involves not just traditional metrics like accuracy and F1-score but also business-driven indicators such as the financial cost of false positives/negatives and operational burden. This aligns model performance more closely with real-world impact.

In conclusion, online banking anomaly detection has evolved into a sophisticated field leveraging diverse techniques across the ML spectrum. Hybrid models, in particular, have demonstrated high adaptability and effectiveness in detecting nuanced fraud behaviors. With ongoing advancements in sequence modeling, graph analytics, and interpretable AI, the future of fraud detection in online banking is poised to become even more robust and intelligent.

2.1.2.2. Review of Domestic Studies

The development of anomaly detection in Vietnam's financial and banking research is a relatively recent phenomenon that emerged primarily in the early to mid-2010s. Prior to this period, Vietnamese academic discourse focused largely on broader topics such as information system security, manual auditing processes, and the infrastructure required for the digital transformation of banking services. To the best of the author's knowledge, based on an extensive manual review of specific indexed Vietnamese academic databases, include the Vietnam Journals Online, Hanoi National University of Education Library Repository, and NEU Scholar from the National Economics University no peer-reviewed publications from the early 2000s explicitly addressed anomaly detection in the context of online banking fraud. The field lacked recognition as a distinct analytical discipline in Vietnam during this time. This reflects a significant research gap in the early phases of digital finance development in the country.

A more discernible shift occurred after 2010, catalyzed by the rapid expansion of digital banking services and the accompanying rise in cyber threats and online financial crimes. Financial institutions including Vietcombank, BIDV, and Techcombank began investing in internal risk management systems and collaborating with domestic technology providers to detect and respond to fraudulent activity when Techcombank, adopted SAS-based fraud detection systems and Vietcombank, VPBank implemented ML-enhanced monitoring tools capable of tracking millions of transactions in real time (Lac Viet, 2023). Additionally, biometric authentication has been deployed to strengthen digital transaction verification; Techcombank now requires facial recognition for online transfers exceeding 10 million VND, and Cake by VPBank has developed its own solution optimized for Vietnamese users (Vietnam Investment Review, 2024). GBG's AI-driven platform Predator has also entered the Vietnamese market, providing solutions based on Random Forest, Neural Networks, and boosting algorithms to financial institutions combating transaction fraud (Vietnam News, 2023). However, specific details and evaluations of these institutional practices are rarely documented in peer-reviewed publications, thus highlighting a limitation in accessible academic literature.

In terms of academic research, related developments have been more commonly observed in the adjacent area of credit card fraud detection, where Vietnamese scholars have begun to experiment with ML-based approaches (Trần & Nguyễn, 2021); however, these studies did not extend to online banking transactions.

Industry partnerships are playing a growing role in narrowing these gaps. Techcombank's integration of SAS fraud detection, Viettel Cyber Security's ML solutions, and strategic insights from firms like KPMG Vietnam have contributed to improved anomaly detection capabilities (Viettel Cyber Security, 2021; KPMG Vietnam, 2025). However, these contributions are typically documented through industry white papers and internal reports rather than formal academic channels, limiting their visibility and impact within the scholarly ecosystem.

In conclusion, despite growing interest in ML and data-driven risk management, the academic literature on anomaly detection in Vietnam's online banking sector remains limited. Most studies focus on accounting or financial statement anomalies, with few addressing transactional fraud using advanced analytics. The scarcity of publicly available datasets, combined with regulatory barriers and a lack of standardized evaluation practices, has hindered broader development. This gap underscores the need for deeper academic-industry collaboration, investment in secure data-sharing frameworks, and adoption of state-of-the-art techniques such as DL and hybrid modeling.

2.2. Proposed Research Model

2.2.1. Research Gaps

Despite the growing sophistication of ML applications in fraud detection, significant research gaps remain, particularly in the context of online banking anomaly detection.

First, although existing synthetic datasets have successfully captured several key features relevant to anomaly detection, they almost focused on card payment and remain isolated from one another and lack standardized, overlapping feature sets. This fragmentation poses a significant limitation, as contemporary online banking systems are simultaneously confronted with multiple forms of financial risk, including transaction fraud, money laundering, and behavioral anomalies, which require a more unified and comprehensive representation of user activity. Furthermore, while there has been notable progress in user profiling within synthetic environments, current approaches remain constrained in their ability to realistically model the amount and temporal dynamics of both inflows and outflows in users' financial behavior. The absence of integrated datasets that capture these dual aspects, including transaction typology overlap and realistic financial timing, hinders the ability to benchmark models under conditions that closely resemble the multifaceted nature of real-world online banking fraud.

Second, the comparative evaluation of anomaly detection models remains fragmented. While DL approaches such as LSTMs, GRUs, and Transformers have shown promise, and hybrid architectures have been proposed, few studies have explicitly benchmarked these models against traditional ML baselines within the same online banking fraud setting using a common dataset and metrics. This lack of holistic evaluation hinders our understanding of when and why one model architecture may be preferable over another, particularly in production settings where false positives carry operational costs and customer experience risks.

2.2.2. Proposed Research Model Methodology

To comprehensively investigate the strengths and limitations of various fraud detection approaches in online banking, this study proposes a framework grounded in both quantitative modeling and qualitative insights. This paper consists of a rule-based simulation model informed by expert-derived heuristics and a three-model framework which enables a systematic comparison between ML, DL, and hybrid approaches using the same synthetic dataset tailored to mimic real transaction behavior. The aim is not only to evaluate performance across models, but also to examine the suitability of each paradigm in capturing contextual and sequential anomalies specific to online banking.

CHAPTER 3: RESEARCH METHODOLOGY

3.1. Data set

3.1.1. Data Source and Description

The dataset utilized in this research was synthetically generated to emulate real-world transactional behavior within online banking systems. Given the confidentiality constraints and limited availability of publicly accessible banking transaction datasets with labeled anomalies, synthetic data generation was employed as a practical and replicable alternative. This approach ensures both the scalability and control necessary to test anomaly detection models under diverse transaction patterns and predefined fraudulent scenarios.

The synthetic dataset simulates transactional records over a period of 24 consecutive months, encompassing approximately 1.5 million transaction entries across unique user accounts within a bank. Each user is assigned a dynamic behavioral profile, including baseline income, typical transaction patterns (frequency, amount, and preferred channels), and seasonal fluctuations. These profiles are designed to reflect heterogeneous financial behaviors observed in actual banking environments, including salary inflows, daily purchases, periodic bill payments, interbank transfers, and discretionary spending.

Each transaction instance contains the following primary features shown in the table below:

Table 3.1: Data Set Attributes Description

Essential Feature	Description
transaction_amount	Monetary value of the transaction. This helps in identifying unusual or suspicious transactions based on deviations from typical account behavior
transaction_type	Indicates whether the transaction is incoming or outgoing. - Incoming Transactions (XXX_TRANSFER): Funds received into the account. - Outgoing Transactions (TRANSFER_XXX): Funds sent from the account. - XXX represents the payment network or

Essential Feature	Description
	bank involved, including NAPAS, CITAD, MB, and VCB.
transaction_date	The date when the transaction occurred (YYYY-MM-DD).
transaction_time	The timestamp indicates when the transaction was executed. This is crucial for detecting patterns like frequent transactions within short intervals.
sender_account	The account number initiating the transaction. If the transaction is outgoing transaction, the sender account belongs to bank user
receiver_account	The account number receiving the transaction. If the transaction is incoming transaction, the sender account belongs to bank user
merchant_id	A unique identifier for the merchant involved in the transaction. Changes in commonly used merchants or high-risk merchant categories can indicate suspicious activity.
trans_channel	Other bank that interacts in the transactions
anomaly_type	A flag indicating whether the transaction is considered anomalous based on predefined rules and detection algorithms. Values may include *Normal* or *Suspicious*.

Source: Author's work

To ensure realism, the data generation process includes both normal and anomalous behaviors, with the latter representing approximately 2% of the total dataset, reflecting real-world class imbalance in fraud detection tasks. All anomalies were generated based on domain-informed rule sets, as discussed in Section 3.1.2.

3.1.2. Rules and Characteristics of Online Banking Transactions

3.1.2.1. Rules for Normal Transactions

(1) Incoming Transactions

Incoming transactions are classified into four main types: (1) Daily Incoming Transactions, (2) Monthly Incoming Transactions, (3) Yearly

Incoming Transactions, and (4) Irregular Transactions. Each type is designed to simulate realistic financial inflows and exhibits distinct characteristics.

Daily Income Transactions

This type of incoming transactions represent frequent, small-scale inflows that occur on a daily basis. They typically simulate regular income sources such as daily wages, small business earnings, or recurring micro-transfers,.... Detail description shown in the table below:

Table 3.2: Daily Income Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	Amounts are generally small to moderate and occur frequently. The transaction values tend to be consistent over time per day, with only slight variations.
transaction_type	Always incoming transactions (XXX_TRANSFER) since they represent income inflows.
transaction_date	Daily transactions with minimal deviation in occurrence, meaning multiple small transactions happen consistently throughout the month.
transaction_time	Consistent transaction times aligned with payroll processing cycles or business hours. Typically occurs during normal hours (5 a.m – 11 p.m). However, 20% of transactions may still occur during midnight to early morning (12 AM – 5 AM) due to job-specific requirements.
sender_account	Random accounts used for transaction generation.
receiver_account	Belongs to individual users within the bank
merchant_id	Randomly selected from the list of merchant_id belong to the user in User table
trans_channel	Randomly selected from the trans_channel in Bank Channels table
anomaly_type	Normal

Source: Author's work

Monthly Income Transactions

Monthly transactions simulate larger, periodic inflows that occur once a month. These transactions often represent salaries, rental income, or subscription-based revenue. Compared to daily transactions, monthly incoming amounts are higher and more predictable. The detail description shown in table below:

Table 3.3: Monthly Income Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	Amounts are moderate to high and occur infrequently (once per month). The transaction values tend to be stable over months, with slight variations due to bonuses, commissions, or deductions.
transaction_type	Always incoming transactions (XXX_TRANSFER) as they represent income inflows. However, since the Sender Account is predetermined, the transaction_type must correspond to each sender_account.
transaction_date	Transactions occur once a month, typically around the 1st - 5th or 25th - 30th of the month, aligning with payroll cycles, rental payment schedules, or subscription payouts. For each individual user, the transaction date falls within a fixed 5-day window and remains consistent across months.
transaction_time	Generally processed during standard business hours (8 AM – 6 PM), aligning with payroll disbursements. However, 10% of transactions may still occur outside regular hours due to automated processing or employer-specific policies.
sender_account	The same sender account tends to repeat over a long period, not changing frequently. Typically represents employers, businesses, or property owners making recurring payments to the receiver.
receiver_account	Belongs to individual users within the bank,

Essential Feature	Characteristics
	representing salary earners, landlords, or subscription service providers.
merchant_id	merchant_id that corresponding to the user merchant
trans_channel	Similar to transaction_type, the trans_channel must correspond to the sender_account, ensuring alignment in the transaction structure.
anomaly_type	Normal

Source: Author's work

Yearly Income Transactions

Yearly transactions represent infrequent but substantial inflows, typically occurring one to four times per year. These transactions commonly include annual bonuses, tax refunds, or lump-sum payments. Due to their high value and low frequency, yearly transactions are particularly sensitive to anomalies, which makes them require careful anomaly detection to prevent fraud or misclassification. The detail description shown in table below:

Table 3.4: Yearly Income Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	Amounts are high and irregular, often significantly larger than daily or monthly income transactions. While values may vary, they tend to remain within a predictable range for each user based on past transactions.
transaction_type	Always incoming transactions (XXX_TRANSFER) as they represent income inflows. However, since the sender_account is predetermined, the transaction_type must correspond to each sender_account.
transaction_date	Typically occurs once to four times per year, often aligning with year-end bonus cycles, tax refund periods, or special company distributions. Each user receives these transactions during a specific and

Essential Feature	Characteristics
	consistent period across different years.
transaction_time	Generally processed during regular business hours (8 AM – 6 PM), aligning with corporate disbursement policies. However, 10% of transactions may occur outside these hours due to automated processing schedules.
sender_account	The sender_account remains highly consistent over time, often associated with corporate entities, government agencies, or financial institutions issuing large payments.
receiver_account	Belongs to individual users within the bank, typically reflecting employees receiving bonuses, taxpayers receiving refunds, or individuals receiving lump-sum payouts.
merchant_id	merchant_id that corresponding to the user merchant
trans_channel	Similar to transaction_type, the trans_channel must correspond to the sender_account, ensuring alignment in the transaction structure.
anomaly_type	Normal

Source: Author's work

Irregular Incomes Transactions

This category encompasses unpredictable and non-recurring inflows that do not follow a fixed temporal pattern. These may include windfalls, gifts, or ad-hoc reimbursements. The irregular nature of these transactions makes them more challenging to model and monitor. The detail description shown in table below:

Table 3.5: Irregular Income Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	Amounts can vary significantly, ranging from small to medium (not the main source of user incoming), depending on the source of income. Unlike regular income transactions, these do not exhibit a consistent pattern over time.
transaction_type	Always incoming transactions (XXX_TRANSFER), as they represent income inflows.
transaction_date	Occurs randomly without a fixed temporal pattern, making it difficult to predict based on historical trends. These transactions may happen multiple times in a year but do not follow a structured recurrence cycle.
transaction_time	Can happen at any time of the day.
sender_account	The sender_account varies widely, as these transactions originate from different individuals, organizations, or financial sources. Unlike salary payments, sender accounts do not follow a consistent pattern.
receiver_account	Belongs to individual users within the bank, representing recipients of unexpected income sources such as gifts, financial assistance, or ad-hoc payments.
merchant_id	merchant_id that corresponding to the user merchant
trans_channel	The trans_channel is selected dynamically, depending on the sender's banking preferences and transaction method. These transactions may be processed through various banking networks without a clear trend.
anomaly_type	Normal

Source: Author's work

(2) Outgoing (Expense) Transactions

Similar to incoming transactions, outgoing transactions are divided into four main types: Daily Outgoing Transactions, Monthly Outgoing Transactions, Yearly Outgoing Transactions, and Irregular Outgoing Transactions. Each type represents different spending behaviors and plays a crucial role in maintaining financial accuracy and anomaly detection.

Daily Outgoing Transactions

Daily Outgoing Transactions represent frequent, small-scale expenditures that occur on a daily basis. These transactions typically cover everyday expenses, including grocery purchases, transportation fees, dining, and minor service payments. While individual transaction amounts tend to be low to moderate, their high frequency makes them a crucial component of personal spending behavior.

Table 3.6: Daily Outgoing Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	Amounts are small to moderate and occur at high frequency. The values remain relatively stable daily, with slight variations based on spending habits.
transaction_type	Always outgoing transactions (TRANSFER_XXX) since they represent expenses. The transaction type depends on the payment method, which varies based on the merchant and service provider.
transaction_date	Transactions occur daily with minimal deviation, meaning multiple small transactions happen consistently throughout the month. Spending patterns may shift slightly on weekends or holidays but generally follow a steady pattern.
transaction_time	Transactions are primarily conducted during active hours (5 AM – 11 PM), aligning with daily shopping and commuting times. However, 20% of transactions may still occur between 12 AM to 6 AM due to late-night purchases, ride-hailing services, or automated payments. This will be checked according to user time behavior in the

Essential Feature	Characteristics
	Users table.
sender_account	The individual user's account within the bank, representing personal expenditures.
receiver_account	Randomly selected from the list of common merchants associated with the user. These often include supermarkets, transportation services, dining establishments, and digital service providers.
merchant_id	Randomly selected from the list of common merchants associated with the user. These often include supermarkets, transportation services, dining establishments, and digital service providers.
trans_channel	Randomly selected from the available payment networks, such as domestic transfer systems, mobile wallets, or direct bank transactions. The channel may depend on user preferences, merchant requirements, or transaction size.
anomaly_type	Normal

Source: Author's work

Monthly Outgoing Transactions

Monthly Outgoing Transactions represent larger, recurring payments that occur once a month. These transactions typically cover essential financial obligations such as rent, utility bills, loan repayments, insurance premiums, or subscription-based services. Unlike daily expenses, monthly transactions are higher in value and more predictable, making them a key component of financial planning. Monthly outgoing transactions are essential for budgeting and financial analysis, as they provide insight into a user's fixed expenses and long-term financial commitments. Their predictability and structured nature distinguish them from other spending patterns, making them crucial in financial behavior modeling.

Table 3.7: Monthly Outgoing Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	Amounts are moderate to high and occur infrequently (once per month). The values remain stable across months, with slight variations due to fluctuations in utility consumption, interest rate adjustments, or changes in service plans.
transaction_type	Always outgoing transactions (TRANSFER_XXX) since they represent expenses. The transaction type depends on the payment method, which varies based on the merchant and service provider.
transaction_date	Transactions typically occur on a fixed date each month, often within a predictable window (usually within 5 days). Each user's transaction date remains consistent across months, aligning with billing cycles, contract terms, or lender policies.
transaction_time	These transactions are usually processed during business hours (8 AM to 6 PM), matching standard billing and payroll operations. However, 10% of transactions may be processed outside regular hours due to automated debits or customer preferences which are checked through user preference in User table
sender_account	The user's personal account within the bank, used to fulfill financial commitments.
receiver_account	Typically remains constant over long periods, as these payments are directed toward landlords, utility companies, financial institutions, or subscription providers.
merchant_id	Corresponds to the specific merchant or service provider receiving the payment. Since recipients are predefined (e.g., landlords, loan providers, subscription services), the merchant_id remains fixed over time.
trans_channel	The payment channel is determined by the type of outgoing transaction. Since recipients are predefined the bank channel remains consistent, ensuring seamless, recurring transactions.
anomaly_type	Normal

Source: Author's work

Yearly Outgoing Transactions

Yearly outgoing transactions refer to large, infrequent payments that occur once or a few times per year. These transactions typically represent long-term financial commitments, such as insurance premiums, property taxes, tuition fees, or membership renewals. Due to their rarity and substantial value, any irregularities in yearly transactions such as unexpected additions, missing payments, or unusually large amounts are easily identifiable and may indicate financial changes or potential errors. Yearly outgoing transactions are critical for financial planning due to their large size and predictable recurrence. Their structured nature makes them ideal for anomaly detection, as missing, unexpected, or excessively high payments can indicate financial issues, errors, or fraudulent activity.

Table 3.8: Yearly Outgoing Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	Amounts are high and occur infrequently (once or a few times per year). The values tend to remain consistent across years, with moderate variations due to price adjustments, inflation, or policy updates.
transaction_type	Always outgoing transactions (TRANSFER_XXX) as they represent expenses. The specific transaction type corresponds to the payment destination (e.g., property taxes are processed through government channels, while insurance payments are routed to financial service providers).
transaction_date	Typically occurs on a fixed annual date or within a predefined window (e.g., January for insurance renewals, April for taxes, or September for tuition payments). Each user's yearly payment schedule remains consistent unless manually adjusted. However, any adjustments are limited to a maximum deviation of one month from the original schedule. As a result, when analyzed over a quarter or half-year period, the transaction pattern remains largely unchanged.

Essential Feature	Characteristics
transaction_time	Transactions are primarily processed during standard business hours (8 am - 6 pm), aligning with financial institution operating hours and billing schedules. However, 10% of transactions will be assigned to be processed outside regular hours due to automated payments or customer preferences.
sender_account	The bank user’s personal or business account used to make annual financial commitments.
receiver_account	Typically remains unchanged over the years, as these payments are directed toward government agencies, educational institutions, insurance companies, or long-term service providers.
merchant_id	Corresponds to the specific organization or entity receiving the payment. Since recipients are predefined (e.g., tax authorities, insurance providers), the merchant_id remains fixed unless a policy or provider changes.
trans_channel	The payment channel is predetermined based on the transaction type. Since recipients are predefined, the bank channel remains consistent, ensuring seamless and timely processing.
anomaly_type	Normal

Source: Author’s work

Irregular Outgoing Transactions

This category includes unpredictable, non-recurring outgoing transactions that do not follow a fixed temporal pattern. Examples include emergency expenses, large one-time purchases, or special occasion spending such as travel bookings, medical bills, or shopping purchases. Due to their sporadic nature, these transactions vary significantly in amount, timing, and frequency, making them more challenging to model and monitor for anomalies.

Table 3.9: Irregular Outgoing Transactions' Characteristics Description

Essential Feature	Characteristics
transaction_amount	The transaction amounts vary significantly, ranging from small incidental expenses to large, unexpected payments. There is no consistency in the amount, as these transactions are event-driven rather than scheduled. However, the transaction amount is limited for user income left after they pay for daily essential and monthly or yearly bills
transaction_type	Always outgoing transactions (TRANSFER_XXX), as these represent expenditures.
transaction_date	Unpredictable and non-recurring, these transactions do not follow a fixed schedule. They can happen 2 to 5 times a month with small transaction amount (eg. random shopping expense) or just 1 to 2 times a year with large amounts (eg.travel booking, luxury items purchased,...)
transaction_time	Transaction times are highly variable. While most occur during regular active hours (5 a.m - 11 p.m), 10% surgent payments may take place during off-hours if related to emergencies or time-sensitive purchases.
sender_account	The sender is always belong to bank user's account
receiver_account	The receiver varies depending on the transaction purpose, including service providers, retailers, healthcare institutions, or individual recipients in cases of peer-to-peer transfers.
merchant_id	Selected randomly from a wide range of merchants, as the nature of irregular expenses is diverse (e.g., travel agencies, hospitals, car dealerships, online marketplaces).
trans_channel	Chosen randomly from the valid trans_channel values
anomaly_type	Normal

Source: Author's work

3.1.2.2 Rules for Anomaly Transactions

Exceed amount of incoming and outgoing transactions

Key Anomaly Detection Criteria:

- Each transaction is compared against the monthly average incoming and outgoing amounts of the respective user. A transaction is flagged as suspicious if the total income or expenditure for the current month exceeds three times that of the previous month (exclude those from yearly income/expense). Even if individual transaction amounts appear normal, their cumulative effect over a month may indicate unusual financial behavior.

Key Attribute to Monitor: Transaction Amount

Repeated Shortly Interval Pattern of Transactions

Key Anomaly Detection Criteria:

- When multiple transactions occur less than three minutes apart, forming patterns such as continuous inflows, continuous outflows, or alternating inflow-outflow cycles. For example, a user may receive multiple small deposits within seconds or transfer funds out in rapid succession, which could indicate fraudulent activity
- Same receiver/sender account within short-time

Key Attribute to Monitor: Transaction Amount, Sender/Receiver Account, Transaction Date, Transaction Time

Smurfing

One of the most common techniques used in money laundering is smurfing, where large sums are broken down into smaller transactions to avoid detection by financial monitoring systems. This anomaly closely resembles the previously discussed pattern of Transfers to Multiple Banks, but with an added layer of complexity designed to avoid detection. In this scenario, transactions originate from or are directed to different banks within the interbank system. This strategy is often employed to obscure the transaction trail, making it more difficult for regulatory authorities to track illicit fund movements, particularly in money laundering schemes.

Key Anomaly Detection Criteria:

- Frequent Small Transactions: Repeated transactions occur just below regulatory reporting thresholds. For example, transactions between 9.8 million and 10 million VND per transaction with a limit of fewer than three transactions per day to evade biometric authentication or other security measures.

- Large Transaction Volume Without Exceeding Limits: Daily transaction amounts are strategically structured between 480 million and 499 million VND, staying just under the threshold that would trigger additional scrutiny.

- Consistent Transaction Flow Patterns: Money laundering transactions often exhibit predictable movement patterns, such as:

➔Continuous incoming transfers (YYY_TRANSFER) from multiple sources.

➔Continuous outgoing transfers (TRANSFER_XXX) to multiple accounts.

➔Immediate withdrawal or transfer out right after receiving funds (YYY_TRANSFER → TRANSFER_XXX), which may indicate layering or integration in the laundering process.

Key attributes to monitor: Transaction Timestamps (to identify rapid or structured patterns), Transaction amounts (to detect threshold-avoidance strategies)

Unusual Merchant Activity

Unusual merchant activity refers to transactions involving a merchant_id that deviates significantly from the account holder's typical spending patterns. While it is normal for users to make transactions to new merchants occasionally, suspicious activity arises when funds are transferred to a new merchant and then, within a very short period (less than three minutes), another transaction occurs at a merchant located in a distant geographical area.

Key Anomaly Detection Criteria:

- Short Interval Between Transaction: Transactions are made to different merchants within an unusually short time frame.

- Geographical Inconsistency: The second merchant transaction is a new merchant which takes place in a significantly distant location from the first merchant. Transactions within a short period occur in different cities or even different countries, which is highly irregular for an individual user.

Key attributes to monitor: Merchant ID, Transaction Time

Transfers to Multiple Banks

This anomaly refers to a pattern where multiple transactions of similar amounts, frequencies, or recipients are dispersed across different banks. This behavior can indicate potential fraudulent activity, money laundering, or attempts to circumvent transaction monitoring systems.

Key Anomaly Detection Criteria:

- Transfers to different banks occur within 3 minutes or even seconds of each other. The rapid execution of multiple transactions suggests an automated or pre-planned operation rather than genuine user behavior.

Key attributes to monitor: Bank Channel, Transaction Time

Dormant Account Activity

Dormant accounts, which have had no activity for an extended period, can become targets for fraudsters.

Key Anomaly Detection Criteria:

- Sudden Large Transactions: A previously inactive account receiving or transferring a significant sum of money.
- Rapid Fund Movements: Transactions occurring in quick succession (e.g., funds credited to an account, followed by an outgoing transfer within three minutes).

Key attributes to monitor: Transaction Time, Transaction Amount

3.1.3. Data Generation Process

The transaction generation process is structured into several distinct phases.

First, three primary tables are generated, containing information about Users, Merchants, and Bank Channels. The Users table includes information about users_id (a unique identifier for each user), user_account (account number), common_merchant_ids (the most frequently used merchant by the user, user_type (categorized as Day and Night users, based on transaction behavior). The Merchant table includes information about merchant_id provided by banks. Bank Channels provides information about the trans_channel of each bank in the Vietnam bank system.

Second, the system generates incoming transactions for each user. These transactions are categorized by frequency and are assigned according to predefined patterns that simulate realistic user behavior.

Third, after establishing the incoming transactions, the system calculates the maximum allowable outgoing transaction limit by multiplying the total incoming amount by 1.2. This calculated limit governs the generation of outgoing transactions, ensuring that users do not exceed a reasonable spending threshold. Once the limits are established, the system proceeds to generate outgoing transactions within the defined boundaries. Each user is assigned a mix of daily, monthly, yearly and irregular outgoing transactions, ensuring diverse transaction activity. The system iterates through all users systematically, repeating this process while maintaining transaction integrity.

Finally, the process moves to the generation of anomalous transactions. These anomalies are created by deliberately introducing irregularities, such as exceeding the transaction limits, manipulating transaction frequency, or creating temporal anomalies. This step enables the system to test and improve its ability to identify suspicious patterns. The process of generating synthetic data shown below:

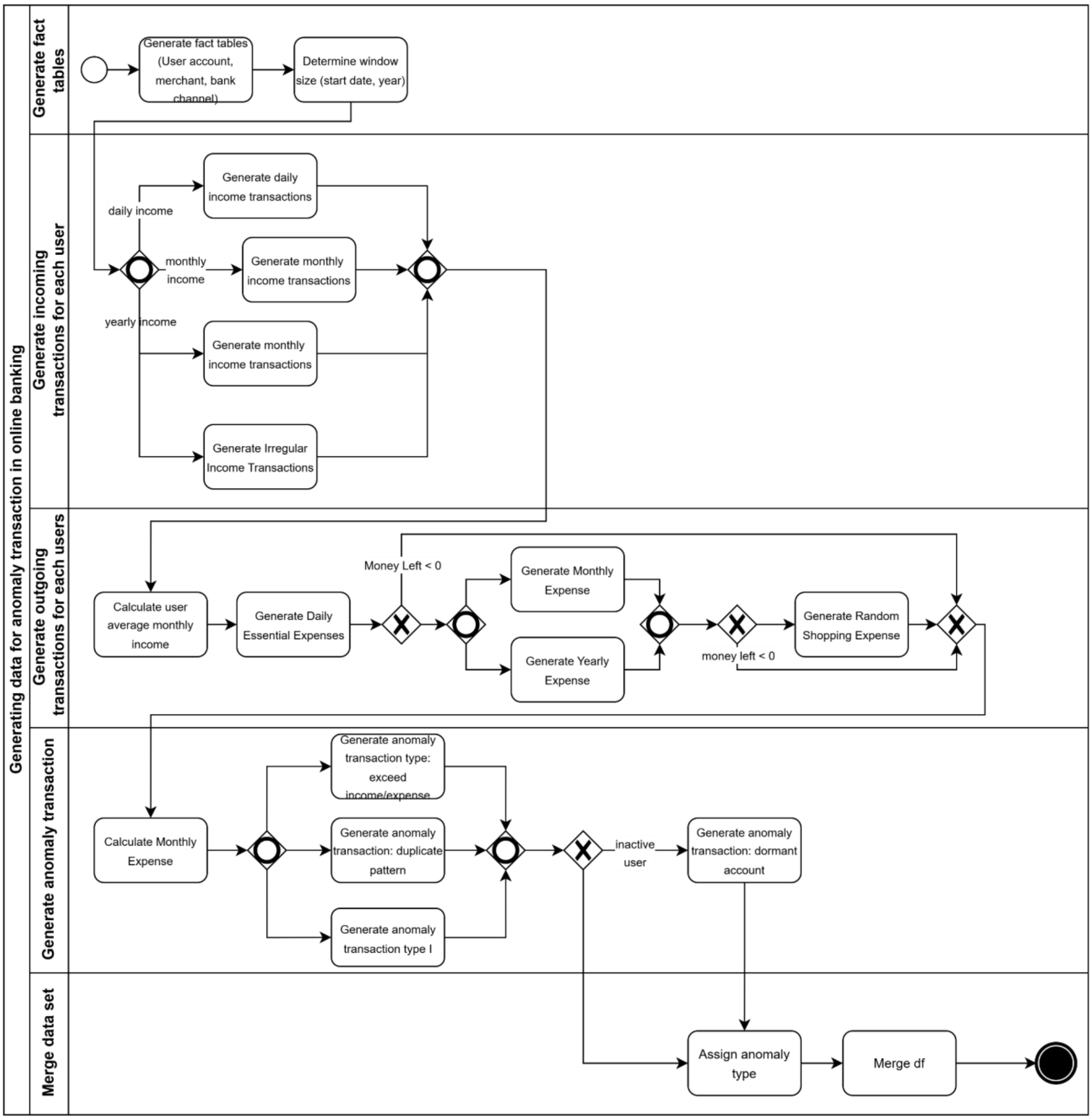
Generating data for anomaly transaction in online banking
Generate fact tables
Generate fact tables (User account, merchant, bank channel)
Determine window size (start date, year)
Generate incoming transactions for each user
daily income
Generate daily income transactions
monthly income
Generate monthly income transactions
yearly income
Generate monthly income transactions
Generate Irregular Income Transactions
Generate outgoing transactions for each users
Calculate user average monthly income
Generate Daily Essential Expenses
Money Left < 0
Generate Monthly Expense
Generate Yearly Expense
money left < 0
Generate Random Shopping Expense
Generate anomaly transaction
Calculate Monthly Expense
Generate anomaly transaction type: exceed income/expense
Generate anomaly transaction: duplicate pattern
Generate anomaly transaction type I
inactive user
Generate anomaly transaction: dormant account
Merge data set
Assign anomaly type
Merge df

Figure 3.1: Data Generation Process

Source: Author's work

3.1.4 Data Processing and Feature Engineering

To support effective model training and enhance the capacity for anomaly detection, a comprehensive data preprocessing and feature engineering pipeline was established. The dataset was transformed into two parallel formats: one optimized for sequential modeling via LSTM, and another for tabular learning via tree-based methods and hybrid classifiers.

3.1.4.1 Sequential Feature Engineering

The following 11 features were selected for use in the LSTM-based sequential model, representing the temporal and contextual characteristics of each transaction. The detail explanation of each feature shown in the table below:

Table 3.10: Sequence Features Description

Feature	Description	Formula / Logic
transaction_amount_log	Log-transformed transaction amount to normalize skewed distributions.	$log(1 + transaction\_amount)$
merchant_id_code	Encoded merchant ID to model merchant behavior patterns.	Integer encoding
day_of_year_scaled	Scaled day of the year to capture seasonal patterns.	$scaled\_day = \frac{day\_of\_year}{365}$
hour	Hour of transaction (0–23), used to model daily	Extracted from timestamp

Feature	Description	Formula / Logic
	rhythm.	
bank_channel	One-hot or label-encoded channel (ATM, online, etc.).	Categorical encoding
other_account	Encoded destination account; important for smurfing/structuring detection.	Integer encoding
is_credit	Binary indicator: 1 = credit, 0 = debit.	$is\_credit \; = \{1, \; if \; inflow \;\; 0, if \; outflow$
is_new_merchant	Merchant not seen in the past 60 days.	$is\_new \; = 1(merchant \; \notin history_{60d})$
amount_near_threshold_flag	Transaction close to known threshold (e.g., 9,900 under 10,000 limit).	
is_first_txn_after_inactivity	First transaction after inactivity window (e.g., >30 days).	$flag = 1(day\_since\_last\_txn \; > 365)$
unusual_merchant_30d	Merchant flagged for abnormally high	Based on system-wide z-score or activity count percentile

Feature	Description	Formula / Logic
	volume/frequency in past month.	

Source: Author's work

3.1.4.2 Tabular Feature Engineering

Another 18 engineered features were used to incorporate aggregated, behavioral, and time-sensitive user signals. The detail explanation of each feature shown in the table below:

Table 3.11: Tabular Features Description

Feature	Description	Formula / Logic
transaction_amount_log	Log-transformed transaction amount to normalize skewed distributions.	$x = log(1 + transaction\_amount)$
is_new_merchant	Merchant not seen in the past 60 days.	$is\_new = 1(merchant \notin history_{60d})$
amount_near_threshold_flag	Transaction close to known threshold (e.g., 9,900 under 10,000 limit).	
is_first_txn_after_inactivity	First transaction after inactivity window (e.g., >30 days).	$flag = 1(day\_since\_last\_txn > 365)$

Feature	**Description**	**Formula / Logic**
rolling_30d_inflow_total	Total inflow for user over the past 30 days.	$\sum_{i=1}^{n} x_i \; where \; x_i > 0 \; over \; 30 \; days$
rolling_30d_outflow_total	Total outflow over past 30 days	$\sum_{i=1}^{n} x_i \; where < 0 \; over \; 30 \; days$
current_outflow_to_12m_avg_ratio	Ratio of this month's income to historical 12-month average.	$r = \frac{income_{current}}{income_{avg}}$
is_new_merchant	Same for outflows.	$r = \frac{outflow_{current}}{outflow_{avg}}$
similar_outflow_spike_in_last_12m	Binary flag if user had similar spike previously.	$1(\exists i \in [1,12] \; s.t. \; x_i >)$
similar_income_spike_in_last_12m	Same for income.	Similar logic
income_spike_this_month	Binary flag for spike beyond certain percentile.	$1(x > \mu + 2\sigma)$
outflow_spike_this_month	Same for outflow.	Same logic
location_flag	Indicates location-based anomaly.	$1(location_{txn} \notin previous\_txn\_location)$

Source: Author's work

3.2. Prediction Model

3.2.1. Machine Learning Algorithm

3.2.1.2. Random Forest Algorithm

The Random Forest (RF) algorithm, introduced by Breiman (2001), is utilized to provide a baseline for ensemble learning methods. A RF consists of a large number of decision trees trained independently on bootstrapped samples of the training data (Breiman, 2001).

A decision tree is a non-parametric supervised learning algorithm used for both classification and regression tasks. It partitions the input space into hierarchical regions by recursively splitting the data along feature thresholds that optimize a particular impurity criterion. For classification, common impurity metrics include:

Gini impurity:

$$G(D) = 1 - \sum_{k=1}^{K} p_k^2$$

or **Entropy**:

$$H(D) = -\sum_{k=1}^{K} p_k(p_k)$$

where p_k is the proportion of instances of class *k* in dataset *D*.

Mathematically, a decision tree seeks to split the dataset *D* into subsets D_{left} and D_{right}at each node *t*, using a threshold θ on feature j, such that a given impurity function (Gini index or entropy) is minimized:

$$j^*, \theta^* = arg\ arg\ G(D_{left}(j,\theta), D_{right}(j,\theta))$$

Where:

G is an impurity measure

j^*, θ^* denote the optimal split feature and value.

This recursive partitioning continues until stopping criteria are met (e.g., maximum depth, minimum samples per leaf), resulting in a tree structure where each path from root to leaf represents a decision rule. While decision trees are interpretable and flexible, they are prone to overfitting and high variance. Ensemble methods like Random Forest, XGBoost, and LightGBM improve upon these limitations by aggregating multiple decision trees in various ways, as detailed in the following subsections.

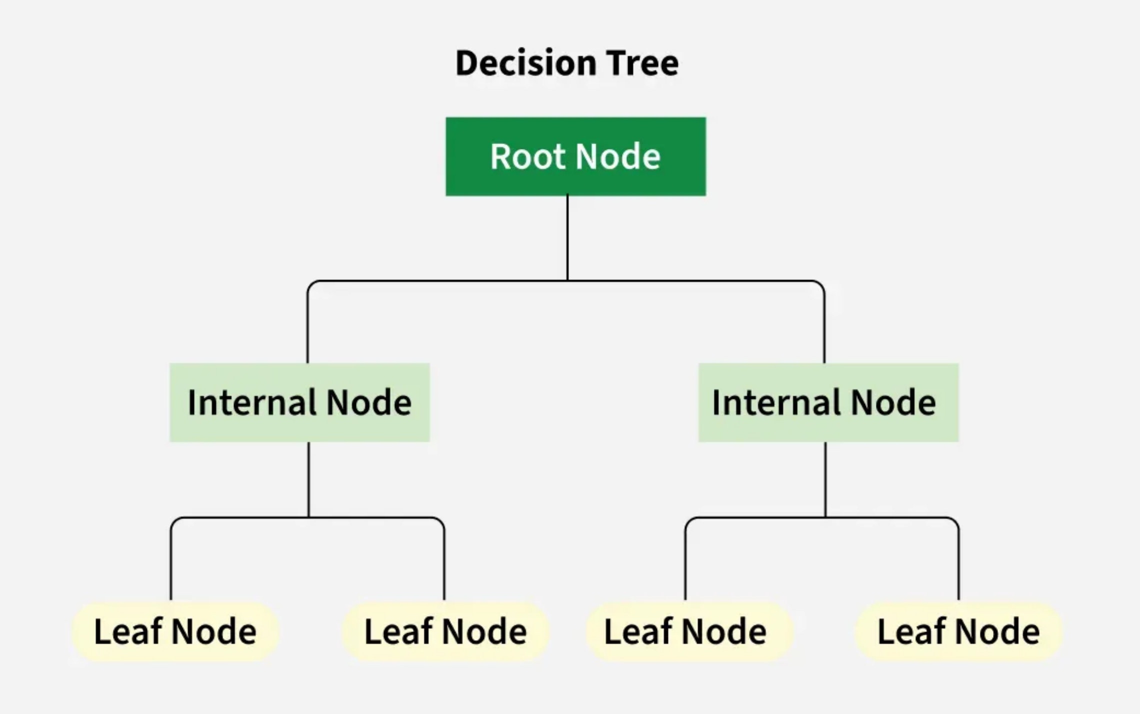

Figure 3.2: Decision Tree Structure

Source: GeeksforGeeks (2025)

In the RF model, each tree makes its prediction by recursively partitioning the feature space, and the final prediction is determined by aggregating the outputs of all trees, typically through majority voting in classification tasks.

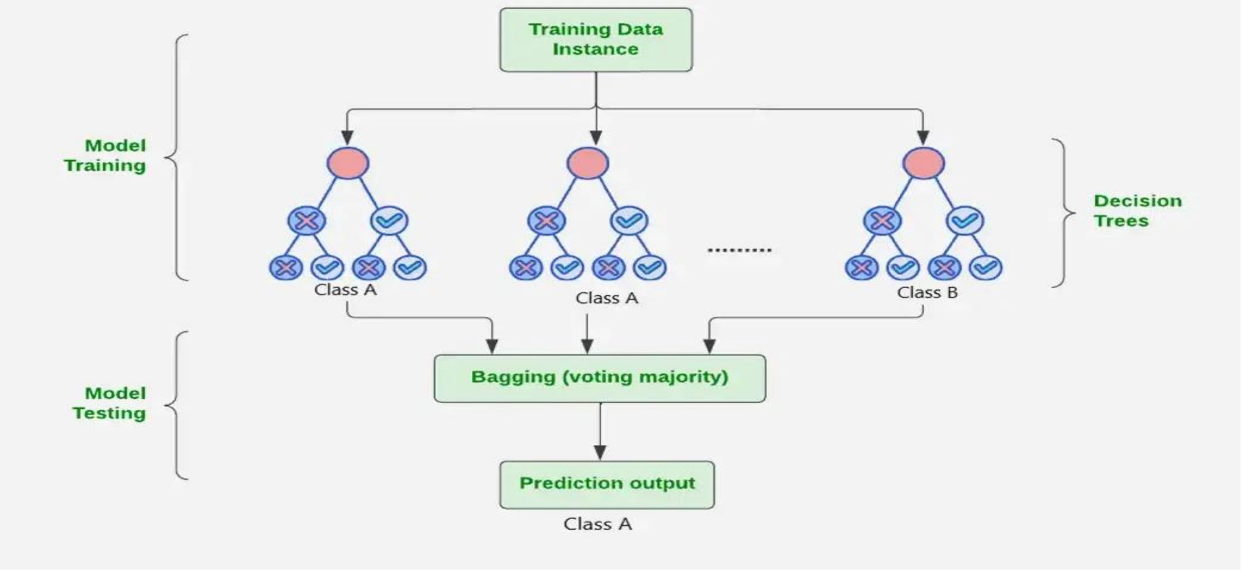

Figure 3.3 : RF Model Structure

Source: GeeksforGeeks (2025)

The prediction for classification tasks is made by majority voting across all trees:

$$\hat{y} = \{mode(h_1(x), h_2(x), \dots, h_m(x)\}$$

In the context of anomaly detection, each transaction is independently evaluated across all trees in the forest. Different subsets of features are considered at each split to introduce additional randomness, thereby reducing overfitting and variance. Transactions that consistently appear in the anomalous class across multiple trees are more confidently classified as anomalies. Due to its parallelizable structure and robust handling of feature correlations, Random Forest provides a strong baseline performance, although it lacks the boosting-based iterative refinement characteristic of XGBoost and LightGBM.

3.2.1.1. XGBoost Algorithm

In this study, the XGBoost algorithm is employed to classify transactions based on a structured set of engineered features. XGBoost is a decision tree ensemble method based on the gradient boosting framework, where new trees are sequentially constructed to correct the residuals (errors) of previous trees. The model aims to minimize a regularized objective function that balances predictive accuracy and model complexity (Chen & Guestrin, 2016). Following the construction of the initial model, the errors or residuals are computed by comparing the predicted values against the actual labels. These residuals represent the component of the data that remains unexplained by the current ensemble. Subsequently, a new decision tree is trained to predict these residuals rather than the original labels. This corrective tree aims to capture the patterns in the data that the previous ensemble failed to model accurately. This aggregation is typically weighted by a learning rate parameter, which controls the contribution of each new tree to prevent overfitting. Mathematically, the model prediction after m iterations can be expressed as:

$$\hat{y}_i = \sum_{k=1}^{m} a_k f_k(x_i)$$

where:

$\hat{y}_i$ is the predicted output for instance i,

a_k is the learning rate associated with the k-th tree,

$f_k(x_i)$ is the output of the k-th tree for input x_i

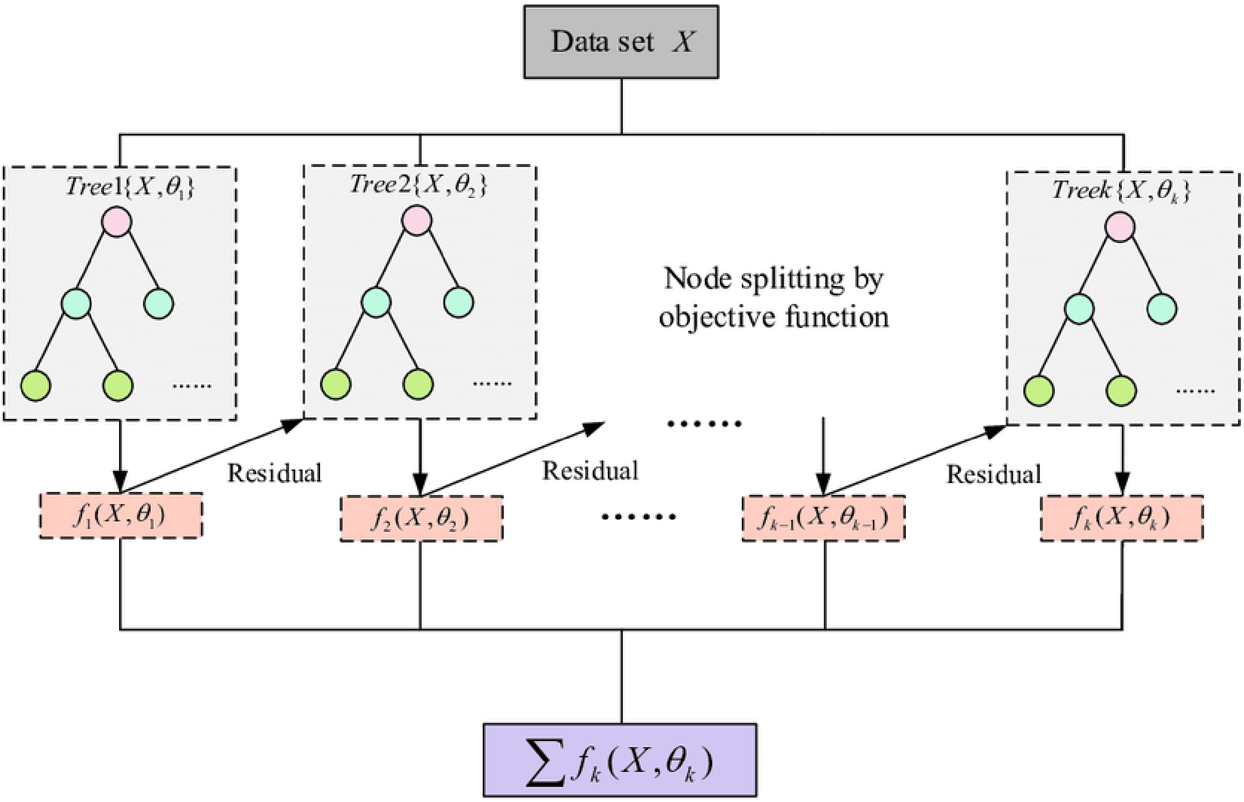

Figure 3.4: XGBoost Model Computing Process

Source: Hidayaturrohman et al. (2024)

Initially, all predictions are made with a simple average or a base score. Residuals, defined as the difference between the actual labels and the predicted probabilities, are computed for each instance. The algorithm then fits a new tree that minimizes the loss associated with these residuals using a second-order Taylor approximation of the loss function. The objective function at the t-th iteration is given by:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i) + \Omega(f_t)$$

where:

l is the loss function (e.g., logistic loss for binary classification),

$\hat{y}_i^{(t-1)}$ is the prediction from the previous iteration,

$\Omega(f_t)$ is the regularization term defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

where T is the number of leaves in the tree, and w_j are the leaf weights. The optimization is performed using a second-order Taylor expansion of the loss function around $\hat{y}_i^{(t-1)}$, leading to efficient approximate updates.

Thus, in the learning process, each new tree attempts to predict the negative gradient (pseudo-residuals) of the loss with respect to current model predictions, and tree structures are grown to minimize this approximation of the loss.

During each iteration, the model evaluates all possible splits across all features and selects the split that maximizes the improvement in the loss function, which is typically the logistic loss in binary classification. The prediction for each transaction is updated by aggregating the output of all previously fitted trees, weighted by a learning rate to prevent overfitting. To control model complexity and improve generalization, regularization terms, including tree depth and minimum child weight, are incorporated into the objective function (Chen & Guestrin, 2016). Thus, in the anomaly detection task, each transaction's tabular features, including recent behavior ratios, transaction amount logs, and merchant activity flags, are analyzed through recursive partitioning to produce a final prediction score indicating the likelihood of being anomalous.

3.2.1.2. LightGBM Algorithm

LightGBM also belongs to the family of gradient boosting frameworks but introduces critical innovations to enhance training efficiency. The core optimization objective of LightGBM Algorithm is similar to XGBoost:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega(f)$$

However, LightGBM determines splits by selecting the leaf that maximizes the gain:

$$Gain = \frac{1}{2}\left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right) - \gamma$$

where G and H denote the sum of first and second-order gradients, respectively, over the left (L) and right (R) partitions

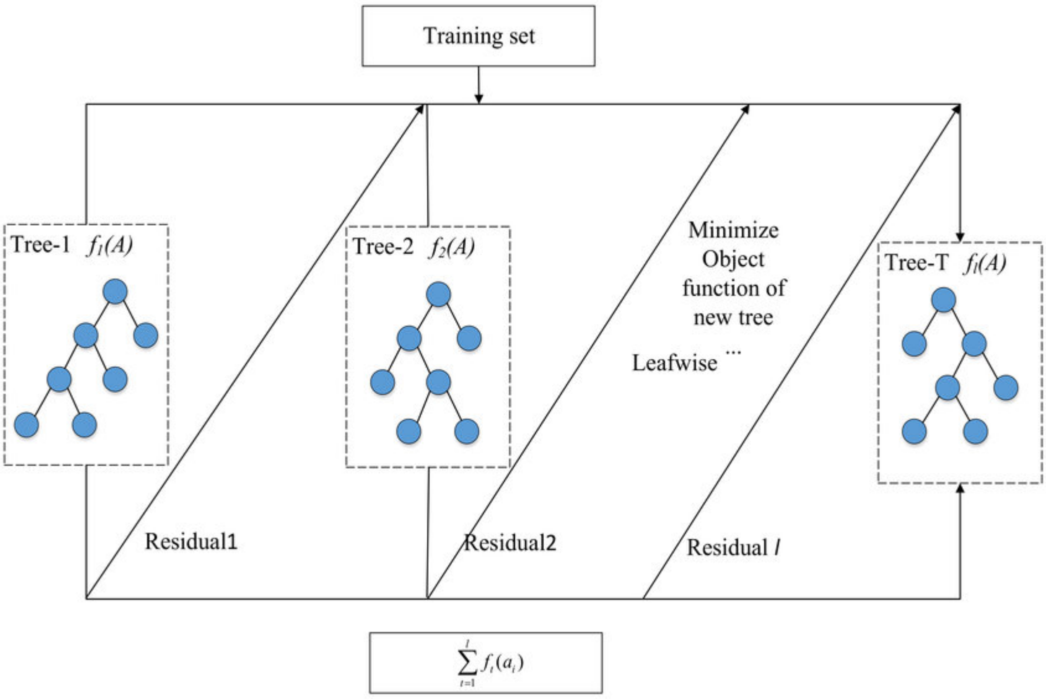

Figure 3.5: LightGBM model computing process

While XGBoost grows trees in a level-wise manner, whereby all leaves at a given depth are expanded before moving to deeper levels. This approach tends to produce more balanced trees and ensures that all parts of the feature space are learned equally. In contrast, LightGBM adopts a leaf-wise tree growth strategy, in which the leaf with the maximum loss reduction is expanded first (Ke et al., 2017). By focusing on the most informative splits, LightGBM typically achieves lower loss values with fewer iterations, resulting in faster convergence. However, this strategy may also lead to more complex and deeper trees, potentially increasing the risk of overfitting if not properly regularized.

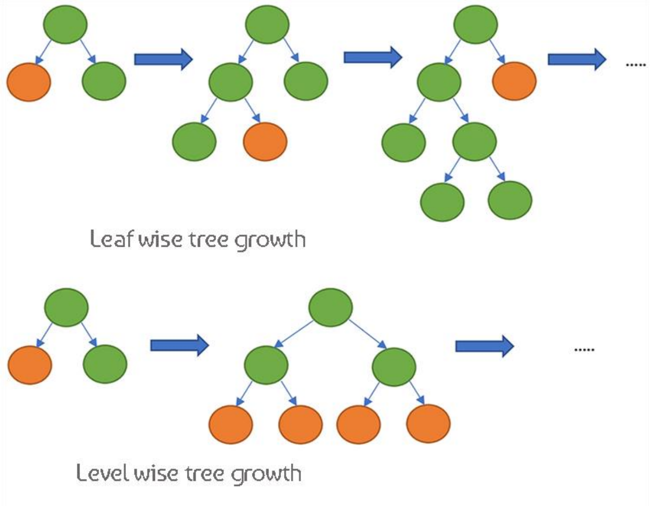

Figure 3.6: Comparison between XGBoost and LightGBM Structure

Source: Mantena et al. (2023)

When applied to transaction data, LightGBM first discretized continuous features into bins to accelerate computation and reduce memory consumption. During model training, it iteratively updates the prediction scores by fitting new trees to the largest-error transactions, thereby refining the distinction between normal and anomalous patterns. Additionally, LightGBM's exclusive feature bundling technique merges sparse features to optimize computation, making it particularly efficient for high-dimensional transaction data. The final prediction is the cumulative output from all trained trees, representing the probability of anomaly occurrence for each transaction.

3.2.2. Long Short-Term Memory Model

The LSTM model is implemented to capture the sequential and temporal dynamics of user transactions. In the context of anomaly detection in online banking transactions, the sequential data consists of time-ordered lists of transactions associated with each user. Each transaction represents a time step in the sequence, characterized by a set of features that describe its contextual, temporal, and behavioral attributes. To construct the input sequences, transactions are first sorted chronologically according to their execution timestamps. For each user, a fixed-length window of the most recent transactions is extracted, forming a transaction sequence. If a user has fewer transactions than the window length, the sequence is padded with special tokens or zero-vectors to achieve uniformity across the batch. Each time step within a sequence is represented by a feature vector comprising information such as log-transformed transaction amount, transaction hour, bank channel, merchant identifier embedding, and binary indicators reflecting behavioral flags like inactivity or new merchant activity.

Formally, the sequence for user can be denoted as:

$$X^{(u)} = [x_1^{(u)}, x_2^{(u)}, x_3^{(u)}, \dots, x_T^{(u)}]$$

where: $x_t^{(u)} \in R^d$ is the feature vector of dimension *d* at time step *t*, and *T* is the sequence length. Each feature vector includes temporal, behavioral, and contextual attributes (e.g., log amount, transaction hour, is_new_merchant, inactivity flag, etc.).

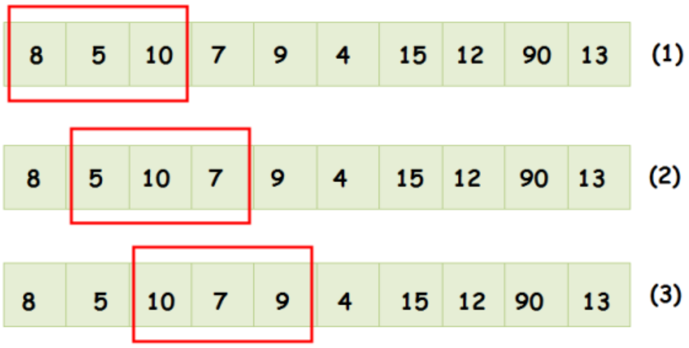

Figure 3.7: Sequence input when sequence_length = 3

Through this mechanism, the LSTM maintains a memory cell C_t that selectively incorporates relevant past information while discarding irrelevant signals, allowing the model to build a dynamic understanding of user behavior over time. After processing the full sequence, the final hidden state h_t serves as the learned representation summarizing the user's recent transactional activity. The LSTM cell maintains a hidden state ht\mathbf{h}_tht and a memory cell ct\mathbf{c}_tct, updated at each timestep via the following equations:

This hidden state is subsequently passed through fully connected layers with sigmoid activation to predict the probability that the final transaction in the sequence is anomalous. Mathematically, the anomaly probability for a sequence is given by:

$$\hat{y} = \sigma(W_{out}h_T + b_{out})$$

where W_{out} and b_{out} denote the parameters of the output layer.

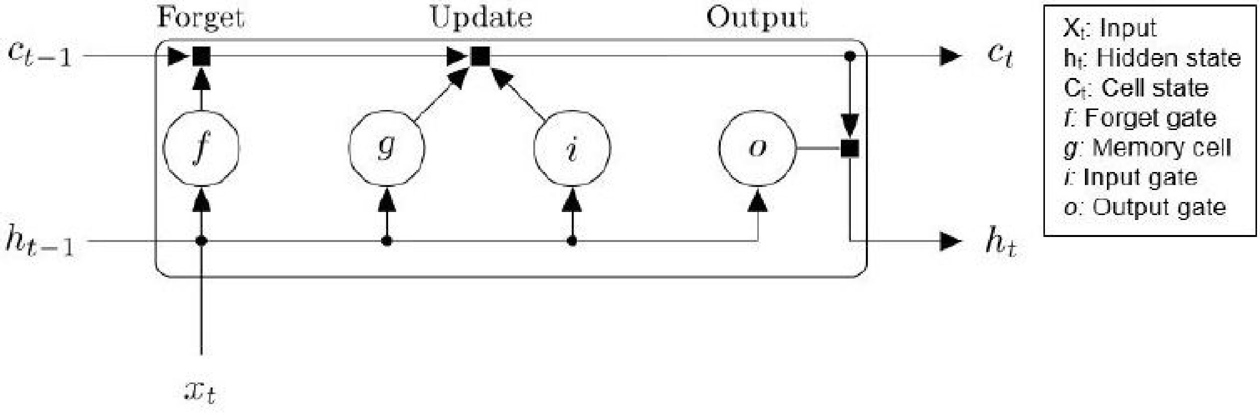

Figure 3.8: LSTM structure

The use of sequential modeling enables the LSTM to detect anomalies not solely based on isolated transaction attributes but by considering the context of preceding user behavior. For example, a transaction with a high amount may not be suspicious if it follows a pattern of similar high-value transactions, but may be anomalous if it follows a long period of inactivity or low-value transactions. This temporal sensitivity makes LSTM architectures particularly effective for fraud detection in dynamic financial environments.

3.2.2. Hybrid Model

To leverage both the sequential learning capacity of LSTM and the structured predictive power of gradient boosting algorithms, a hybrid model is constructed. The hybrid modeling process begins with the LSTM processing each user's transaction sequence, as described previously. The output of the LSTM, specifically the final hidden state of each sequence, is extracted as a compact, learned representation of recent behavioral patterns.

This LSTM embedding is then concatenated with the original tabular features engineered from the transaction dataset. Thus, each transaction is now represented by a feature vector that contains both static context (e.g., rolling transaction averages, month cyclic encodings) and dynamic sequential behavior learned from transaction history.

$$z = [h_{LSTM}; x_{tabular}]$$

This combined vector z is then fed into a traditional ML classifier, which analyzes both sequential dynamics and static transaction context to predict anomaly probabilities.

3.3. Evaluation Metrics

Given the inherent challenges of detecting anomalies in online banking transactions, particularly the severe class imbalance between normal and anomalous instances, the evaluation of predictive models requires a comprehensive set of performance metrics. A single metric is insufficient to fully characterize model behavior under these circumstances. Consequently, this study employs a combination of classification metrics and curve-based evaluation techniques, following the recommendations of Ahmed, Mahmood, and Hu (2016) and Saito and Rehmsmeier (2015).

The first and most basic metric considered is accuracy, which measures the proportion of correctly predicted instances over the total number of instances. The formula for accuracy is expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP denotes true positives, TN true negatives, FP false positives, and FN false negatives. While accuracy is intuitive, it becomes unreliable in highly imbalanced datasets, as a model can achieve high accuracy simply by predicting the majority class without correctly identifying any anomalies. Therefore, greater emphasis is placed on precision, recall, and the F1 score. Precision is defined as the proportion of correctly identified anomalies among all instances predicted as anomalies, capturing the model's ability to avoid false positives. Mathematically, precision is given by:

$$Precision = \frac{TP}{TP + FP}$$

Recall, alternatively known as sensitivity or the true positive rate, measures the proportion of actual anomalies that are correctly detected by the model. It is formulated as:

$$Recall = \frac{TP}{TP + FN}$$

The F1 score, being the harmonic mean of precision and recall, provides a balanced evaluation metric that considers both false positives and false negatives. It is calculated as:

$$Recall = 2 * \frac{Precision * Recall}{Precision + Recall}$$

CHAPTER 4: EMPIRICAL RESULTS

4.1. Data Generation Results

The experimental evaluation in this study is conducted using the synthetic transaction dataset developed according to the rules, structures, and anomaly injection processes detailed in Section 3.1. The dataset was generated to closely simulate real-world online banking transactional behavior over a period of 24 consecutive months, encompassing approximately 1.5 million transaction records distributed across 50 user profiles.

The dataset reflects a diverse mixture of daily, monthly, yearly, and irregular financial activities on both the income and expenditure sides, aligned with the behavioral models established during the synthetic generation phase. In addition to primary transaction features, multiple engineered variables capturing rolling behavior statistics, cyclic time encodings, and anomaly context flags were introduced to enhance the model's ability to detect complex fraudulent patterns.

Anomalous transactions account for approximately 1% of the entire dataset, mirroring the class imbalance typically observed in real-world financial systems. The anomalies encompass a variety of patterns, including sudden outflow spikes, repeated short-interval transactions, structured money laundering flows, unusual merchant activities, and dormant account reactivations, as outlined in Section 3.1.2.2.

For the purposes of model evaluation, the dataset was partitioned into non-overlapping training and testing sets. Temporal ordering was preserved to simulate a realistic predictive environment where models are trained on past transactions and evaluated on future unseen data. Stratification by user accounts was also applied to prevent information leakage, ensuring that each user appears exclusively in either the training or testing partition. The final prepared dataset thus provides a comprehensive experimental ground for testing the effectiveness of ML, sequential DL, and hybrid anomaly detection models under conditions approximating actual online banking environments.

4.2. Model Performance Results

4.2.1. Machine Learning Models

The first group of models evaluated consists of three widely used ML algorithms: XGBoost, LightGBM, and Random Forest. These models were trained using the tabular features described in Section 3.1.4.2, with hyperparameters tuned through randomized search with cross-validation to optimize performance.

The performance metrics for each ML model on the test set are summarized in Table

Table 4.1: ML Models results

Model	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
XGBoost	0.7617	**0.9974**	0.8638	1.0000
LightGBM	0.0607	0.9872	0.1143	0.9551
RF	**0.9392**	0.9456	0.9424	0.9999

Source: Author's work

The Random Forest model achieved the most balanced results among all ML models tested. It recorded a precision of 0.9392, meaning that over 93% of flagged transactions were truly fraudulent, and a recall of 0.9456, indicating that 94.56% of all real fraud cases were captured. The high F1 score of 0.9424 confirms that Random Forest maintained an excellent balance between minimizing false positives and capturing true anomalies, achieving both high detection rates and high prediction quality. The ROC AUC score of 0.9999 also demonstrates that Random Forest was highly capable of distinguishing between classes at varying thresholds, providing flexibility in setting operating points based on business requirements.

Meanwhile, The LightGBM model displayed a notably different behavior. Although it achieved an extremely high recall of 0.9872, meaning that almost all fraudulent transactions were detected, its precision dropped sharply to 0.0607. This means that only around 6% of the flagged transactions were actually fraudulent, while the rest were false alarms. Such low precision is problematic in operational environments, as it could overwhelm fraud analysts with large numbers of false positives, thereby reducing system efficiency and trustworthiness. The F1 score of LightGBM, at only 0.1143, reflects the imbalance between precision and recall. Despite its success in detecting anomalies (high recall), the inability to maintain reasonable precision led to poor overall performance in terms of practical deployment. The ROC AUC score of

0.9551 remains relatively high, suggesting that in theory, LightGBM can distinguish anomalies from normal transactions; however, its practical effectiveness depends heavily on threshold tuning and further calibration, particularly under conditions of severe class imbalance.

The XGBoost model achieved strong overall performance. It obtained a precision of 0.7617, meaning that approximately 76% of transactions flagged as anomalies were truly anomalous. Precision reflects the model's ability to minimize false alarms, a critical factor in maintaining trust in fraud detection systems. Moreover, XGBoost demonstrated a very high recall of 0.9974, indicating that nearly all true fraudulent transactions were successfully identified. The resulting F1 score of 0.8638, which harmonizes precision and recall into a single measure, confirms that XGBoost also achieved a strong balance between capturing fraudulent activities and minimizing unnecessary interventions; however, this number is still lower in the comparison with RF model. In terms of ranking ability, XGBoost achieved a perfect ROC AUC of 1.0000, implying that the model can completely separate anomalies from normal transactions across different decision thresholds.

The confusion matrix (Figure 4.1) shows that only three anomalies were missed, and 367 normal transactions were incorrectly flagged. Given the context of fraud detection, this represents a favorable trade-off where the system prioritizes catching fraud at the slight cost of raising a manageable number of false alerts.

4.2.2. LSTM Model Results

The LSTM model was evaluated based on its ability to capture temporal patterns in sequential transaction data. Unlike traditional ML models that treat transactions independently, the LSTM model processes sequences of historical transactions for each user, learning behavioral patterns over time to predict the likelihood of an anomaly occurring at the end of the sequence.

Table 4.2: LSTM model results

Model	Data
Accuracy	0.9956
Precision	0.5042
Recall	0.7168
F1 Score	0.5920
ROC AUC Score	0.9098

Source: Author’s work

The LSTM model achieved a precision of 0.5042, meaning that approximately 50% of the transactions flagged as anomalies were truly anomalous. Precision reflects the model's ability to minimize false positives that is, to avoid raising unnecessary fraud alarms. While a precision of 0.5 is moderate, it indicates that the LSTM model produces a significant number of false positives, which could burden fraud monitoring teams if deployed without post-processing.

In contrast, the LSTM attained a recall of 0.7168, capturing around 72% of all true anomalous transactions. This score reflects the model's moderate success in detecting fraudulent behavior, although it leaves a notable proportion (approximately 28%) of anomalies undetected.

The resulting F1 score of 0.5920, which balances precision and recall, reflects the model’s intermediate performance. The F1 score, being the harmonic mean, penalizes significant discrepancies between precision and recall, explaining why the moderate precision and recall produce a moderate F1 value. In terms of overall ranking ability, the LSTM achieved a ROC AUC score of 0.9098. This suggests that the model possesses good, though not perfect discriminative ability between normal and anomalous transaction sequences across various decision thresholds. Compared to the ML models evaluated in Section 4.2.1, the LSTM’s ROC AUC is slightly lower, indicating some limitations in its ability to distinguish fraudulent patterns based purely on sequence dynamics.

The moderate precision and recall achieved by the LSTM model can be attributed to the complexity and variability of individual transaction sequences. In many cases, legitimate users exhibit diverse spending behaviors that may resemble fraud patterns when viewed over short time horizons, leading to false positives. Conversely, sophisticated fraudulent behaviors designed to mimic legitimate transaction patterns can evade sequential detection models, contributing to false negatives.

4.2.3. Hybrid Model Results

The hybrid model in this study was constructed by combining sequential transaction embeddings learned via the LSTM network with the engineered tabular features described in Section 3.1.4.2. By leveraging both temporal transaction patterns and static behavioral attributes, the hybrid approach aims to enhance anomaly detection performance beyond what either modality can achieve independently.

Table 4.3: Hybrid Model Results

Metrics	Data
Accuracy	0.9995
Precision	0.8216
Recall	0.9983
F1 Score	0.9013
ROC AUC Score	0.9999

Source: Author's work

The hybrid model achieved a precision of 0.8216, indicating that approximately 82% of flagged transactions were truly fraudulent. The model's recall reached an exceptional 0.9983, meaning that nearly all fraudulent transactions were successfully detected. The resulting F1 score of 0.9013 reflects a strong balance between precision and recall. The F1 score harmonizes these two metrics and is particularly important when both false positives and false negatives carry significant operational consequences, as is the case in banking fraud detection. In terms of ranking capacity, the hybrid model achieved a ROC AUC score of 0.9999, indicating near-perfect ability to discriminate between normal and anomalous transactions across various threshold settings. This performance suggests that the hybrid model can adaptively balance fraud detection and customer experience across different risk tolerance levels.

4.3. Comparative Analysis

4.3.1. Performance Comparison Across Models

The comparative evaluation across XGBoost, LightGBM, Random Forest, LSTM, and Hybrid models reveals significant differences in their abilities to detect anomalous banking transactions.

Table 4.4: Comparison Performance between Prediction Model

Model	Precision	Recall	F1 Score	ROC AUC
XGBoost	0.7617	0.9974	0.8638	1.0000
LightGBM	0.0607	0.9872	0.1143	0.9551
Random Forest	0.9392	0.9456	0.9424	0.9999
LSTM	0.5042	0.7168	0.5920	0.9098
Hybrid	0.8216	0.9983	0.9013	0.9999

Source: Author's work

Among the traditional ML models, Random Forest achieved the highest F1 score (0.9424), reflecting an excellent balance between precision (0.9392) and recall (0.9456). XGBoost also performed strongly, particularly in recall (0.9974), though its slightly lower precision (0.7617) compared to Random Forest led to a lower F1 score (0.8638). LightGBM, in contrast, while achieving very high recall (0.9872), exhibited extremely low precision (0.0607), resulting in the lowest F1 score (0.1143) among all models evaluated.

The LSTM model, leveraging sequential transaction data, achieved a moderate F1 score of 0.5920, driven by a precision of 0.5042 and a recall of 0.7168. While its ability to capture temporal dependencies offered some advantage over simpler tabular models, the overall performance lagged behind Random Forest and XGBoost.

The hybrid model's substantial improvement over the standalone LSTM and ML models can be attributed to its ability to jointly leverage dynamic sequence-based insights and static behavioral features. While the LSTM component captures evolving user behavior and short-term anomalies, the tabular features encode long-term aggregate behaviors and structural financial indicators, such as rolling averages and periodic deviations. By combining these two information sources, the hybrid model mitigates the weaknesses of each individual approach, notably, the LSTM’s moderate precision and the tabular models’ limited temporal sensitivity. This architectural synergy allows the hybrid model to not only detect subtle anomalies that manifest across transaction histories but also to avoid overreacting to transient irregularities that may not signify actual fraud.

However, it is worth noting that the hybrid model outperformed all other approaches except Random Forest in raw F1 score, achieving 0.9013, a precision of 0.8216, and an almost perfect recall of 0.9983. Its ROC AUC of 0.9999 confirmed that the hybrid model could distinguish normal and anomalous transactions with high fidelity across various decision thresholds. By integrating both sequential and static features, the hybrid model demonstrated a significant advantage in real-world fraud detection scenarios where both behavioral patterns and transaction contexts are informative.

Another problem rise in the increased complexity of the hybrid model introduces additional challenges, including greater computational demands, more complex model maintenance, and potentially longer inference latency. Therefore, while the hybrid model achieves superior predictive performance, its adoption must consider the operational constraints and real-time requirements of the intended deployment environment.

4.3.2. Analysis under Imbalanced Data Conditions

The anomaly detection problem addressed in this study is characterized by an extremely imbalanced data distribution, with fraudulent transactions constituting under 1% of the dataset. Under such conditions, traditional metrics like accuracy become misleading, necessitating a focus on precision, recall, F1 score, and PR AUC to meaningfully assess model performance (Saito & Rehmsmeier, 2015).

LightGBM's behavior under imbalance highlights the dangers of relying solely on recall: while the model correctly identified almost all anomalies, its indiscriminate labeling of normal transactions as anomalies produced an unacceptably low precision, overwhelming the system with false alarms. This confirms that models maximizing recall without regard to precision are ill-suited for operational deployment in imbalanced anomaly detection tasks.

The LSTM model, by contrast, achieved a more balanced performance. Its sequence modeling capabilities allowed it to detect anomalies that exhibit temporal dependencies. However, due to the noisiness and variability of real transaction sequences, its precision remained moderate, resulting in an overall F1 score lower than the tree-based models.

XGBoost and Random Forest handled class imbalance particularly well. Both models achieved high recall and high precision simultaneously, illustrating the ability of gradient boosting and bagging ensembles to adapt decision boundaries to capture rare classes without significant overfitting. The use of scale_pos_weight in XGBoost and class_weight adjustments in Random Forest were critical in achieving this performance.

The hybrid model demonstrated the best handling of imbalance overall, achieving extremely high recall (0.9983) while maintaining strong precision (0.8216). Although Random Forest achieved higher precision (0.9392) than the hybrid model, the hybrid model's near-perfect recall implies that almost no anomalies were missed. In practical banking fraud detection, missing fraudulent transactions (false negatives) carries significantly higher costs than investigating additional false positives. Consequently, a model that achieves nearly perfect detection of frauds while sustaining acceptable levels of false alarms is preferable. The hybrid model, by optimizing this trade-off, ensures comprehensive fraud capture without overwhelming operational processes, thereby offering superior real-world performance under imbalanced conditions.

CHAPTER 5: DISCUSSION AND IMPLICATIONS

5.1. Research Implications

5.1.1. Theoretical Implications

This study provides several important theoretical and practical contributions, stemming from the research gaps identified in earlier sections and confirmed through the empirical results of model comparisons. These contributions revolve around three main aspects: the method of generating data for anomaly detection, the construction of rules for anomaly identification, and the comparative evaluation of traditional ML, DL, and hybrid models.

First, although there has been a growing body of international studies on fraud detection using ML and DL approaches, the number of works specifically addressing the challenge of synthetic data generation for online banking transactions remains limited. Moreover, previous synthetic datasets often lacked behavioral complexity, focusing primarily on isolated statistical properties rather than replicating full user transaction profiles over time. Recognizing this gap, the present research developed a synthetic data generation method that modeled diverse banking behaviors across multiple months, including salary inflows, discretionary spending, interbank transfers, and fraud typologies. By constructing a dynamic, behaviorally consistent dataset, the research provides a theoretical foundation for future studies aiming to benchmark anomaly detection models under realistic but privacy-preserving conditions. This represents a meaningful theoretical advance because, particularly within the financial sector, access to labeled real-world transaction data is often severely restricted due to privacy and regulatory concerns.

Second, while previous research has extensively explored various anomaly detection models, there has been a lack of systematic frameworks for defining behavioral rules for anomalies within synthetic banking datasets. Most studies either relied on arbitrary statistical outlier assumptions or vague anomaly criteria, which limited their applicability to real-world fraud detection. In response, this research formulated a set of explicit, behaviorally grounded rules reflecting common fraud patterns: transactions exceeding income thresholds, repeated short-interval transfers suggestive of layering or smurfing, and sudden activity from dormant accounts. By embedding these domain-informed rules into the labeling process, the study enhances the interpretability and theoretical validity of anomaly detection tasks. This approach can serve as a benchmark for future work seeking to align anomaly labeling more closely with actual financial risk patterns rather than purely mathematical deviations.

Third, the comparative evaluation conducted between traditional ML models, DL architectures, and hybrid systems addresses another important research gap in Vietnam. Previous studies often focused on single-model performances in isolation without systematically examining their strengths and weaknesses under a unified experimental environment. The findings of this research demonstrate that while traditional ensemble models such as XGBoost perform robustly on engineered features, and LSTM models excel at capturing sequential dependencies, hybrid models that integrate both approaches substantially outperform either category alone.

Thus, the theoretical contributions of this research not only extend knowledge about synthetic data generation and anomaly labeling in banking but also enrich the understanding of hybrid modeling strategies for complex anomaly detection problems. These findings provide a foundational reference point for future investigations seeking to design more effective, interpretable, and adaptive fraud detection systems across financial and other high-risk domains.

5.1.1. Practical Implications

From the research results and theoretical contributions outlined above, several practical implications are proposed for various stakeholders, including consumers, financial institutions, fintech developers and regulatory bodies.

For *consumers*, although the technical aspects of fraud detection are often invisible, improved anomaly detection systems offer indirect but important benefits. Enhanced anomaly monitoring based on hybrid models can lead to faster fraud identification, reduced financial losses, and greater confidence in digital banking platforms. However, consumers must also be aware of potential limitations, such as the possibility of occasional false positives, and maintain vigilance by adopting good security practices, including monitoring transaction alerts and safeguarding account credentials.

In relation to *financial institutions*, the findings suggest that banks should consider upgrading their fraud detection systems by integrating hybrid models capable of capturing both static transactional features and dynamic sequential behaviors. The demonstrated superiority of the hybrid model indicates that such systems could significantly reduce false positives while maintaining high fraud detection rates. For operational teams, adopting adaptive fraud monitoring pipelines based on hybrid architectures could not only improve security but also enhance customer trust by minimizing unnecessary transaction disruptions.

Regarding *technology developers and fintech companies*, the study highlights the importance of designing modular, explainable, and scalable fraud detection solutions. As hybrid architectures introduce greater complexity, fintech innovators must prioritize the integration of explainability tools, such as SHAP analysis or attention mechanisms, to ensure that fraud detection outputs can be audited and interpreted by human analysts. This is crucial not only for internal validation but also for meeting emerging regulatory expectations concerning algorithmic transparency in financial services.

5.2. Research Recommendation

The first recommendation concerns the *enhancement of regulatory frameworks for AI-based fraud detection.* As online banking transactions increasingly rely on ML and hybrid models for real-time anomaly detection, existing legal and regulatory structures must evolve to accommodate these technological advancements. Financial regulatory bodies should consider developing clear guidelines on the ethical use of artificial intelligence in fraud detection, emphasizing standards for data privacy, algorithmic transparency, model validation, and consumer rights protection. Particularly in jurisdictions where financial technologies have outpaced regulatory oversight, such initiatives are essential to balance innovation with accountability. Clearer rules surrounding explainability requirements, auditability of detection models, and consumer dispute resolution mechanisms would provide a foundation for safer, more reliable AI deployment in banking operations.

Second, it is recommended that governments and financial authorities actively *support the creation of privacy-preserving data sharing mechanisms* to accelerate research and development in fraud detection. One major limitation highlighted in this study was the reliance on synthetic data due to the inaccessibility of real-world banking datasets. Initiatives that promote secure, anonymized data sharing between banks, research institutions, and technology developers could significantly enhance the ability to validate and refine fraud detection models under realistic conditions. For instance, frameworks based on federated learning or synthetic data validation partnerships can ensure that sensitive customer information remains protected while enabling broader collaborative innovation. Establishing national or regional "data trusts" specifically for financial research purposes could serve as a practical model for balancing security and progress in this critical area.

The third recommendation focuses on encouraging the banking sector and fintech industry to *invest in technological innovation for real-time, adaptive fraud monitoring systems.* The empirical findings of this research demonstrated the superiority of hybrid models, such as LSTM-XGBoost architectures, in capturing complex transaction patterns and improving detection accuracy. To translate these findings into practice, banks should prioritize upgrading their fraud detection infrastructures to incorporate advanced ML pipelines capable of learning temporal patterns and adapting to evolving fraud strategies. Investment in dynamic feature engineering, continuous model retraining, and low-latency model deployment platforms would significantly enhance the resilience of financial systems to emerging threats. Moreover, fintech innovators should be incentivized to develop modular fraud detection solutions that integrate easily into diverse banking environments, ensuring broader access across institutions of varying sizes and technological maturity.

Finally, it is recommended that *cross-sector collaboration between government agencies, financial institutions, academic researchers, and technology developers* be strengthened to continuously address the evolving challenges of financial fraud. Given the adaptive nature of fraudsters and the rapid pace of technological change, isolated efforts are unlikely to sustain long-term success. National strategies that foster interdisciplinary research programs, public-private innovation labs, and regulatory sandboxes for safe AI experimentation could greatly accelerate the deployment of effective, ethical, and inclusive fraud detection systems. Furthermore, regular forums and working groups focused on sharing best practices, emerging threat intelligence, and research findings would promote collective resilience in the face of increasingly sophisticated financial crimes.

In conclusion, the research underscores the urgent need for a coordinated, forward-looking approach to anomaly detection in online banking. By enhancing regulatory frameworks, facilitating ethical data sharing, investing in adaptive technologies, prioritizing explainability, and fostering cross-sector collaboration, stakeholders can build a safer, more innovative, and more trustworthy financial ecosystem for the digital era.

5.3. Limitations and Future Research

5.3.1. Research Limitations

Despite the contributions outlined, certain limitations inevitably accompany the research process due to constraints in data availability, resource allocation, modeling scope, and temporal assumptions. Recognizing these limitations is crucial for contextualizing the findings and identifying opportunities for further advancement.

The *first* limitation stems from the use of synthetic data to simulate banking transactions. Although the dataset was carefully constructed to reflect realistic user behaviors, including salary inflows, discretionary spending, and structured fraud scenarios, synthetic data inherently lacks the full complexity and unpredictability observed in real-world financial systems. Fraudsters continuously adapt to circumvent detection, often exhibiting creative patterns that are difficult to anticipate or embed within simulated environments. As a result, while the dataset provides a strong basis for controlled experimentation, it may not fully capture the adversarial dynamics encountered in live banking operations.

Secondly, while the anomaly labeling framework incorporates multiple known fraud patterns, such as exceeding income thresholds, layering, and account reactivation anomalies, the taxonomy of fraud behaviors continues to expand. Emerging threats, such as synthetic identity fraud or real-time social engineering attacks, were beyond the scope of the current rule set. This limitation highlights the inherent trade-off between defining clear, interpretable labeling criteria and fully capturing the evolving landscape of financial anomalies.

A *third* constraint involves the evaluation setting, which was limited to a single synthetic banking institution profile. Although the transaction profiles were designed to be broadly representative, real-world variability across banks, regions, and customer demographics could influence model performance. The absence of multi-institutional or cross-regional testing restricts the generalizability of the conclusions and raises questions about how well the findings would transfer to other banking ecosystems.

The *fourth* limitation pertains to the absence of temporal fraud evolution (concept drift) within the dataset. In operational environments, fraud patterns shift over time as perpetrators adapt to detection strategies. By conducting experiments within a temporally static dataset, the research ensured controlled comparisons between models but did not evaluate the capacity of detection systems to maintain performance over long-term deployments under changing conditions.

5.3.2. Proposed Improvements

Building upon the limitations identified, several avenues for future research are proposed to extend, refine, and validate the findings presented. These directions aim not only to address current constraints but also to enhance the practical relevance, adaptability, and robustness of anomaly detection systems for online banking.

First, validating models against anonymized real-world transaction datasets remains a crucial priority. While synthetic data provides an invaluable tool for initial experimentation, access to real banking datasets, through privacy-preserving partnerships or participation in initiatives like IEEE-CIS Fraud Detection, would enable deeper evaluation of model effectiveness in uncontrolled environments. Such validation efforts would also allow testing under more diverse fraud conditions and transaction behaviors reflective of different customer bases.

Second, future work should expand the definition and categorization of anomalies beyond the patterns captured in the current rule framework. Incorporating subtle or emergent fraud behaviors, such as synthetic identity creation, cross-border money movement anomalies, or behavioral biometrics manipulation, would enrich the realism of training and evaluation datasets. These expansions could be guided by collaboration with fraud investigation units to continually integrate field observations into synthetic generation pipelines.

Third, addressing the issue of concept drift is critical for designing resilient fraud detection systems. Future datasets should incorporate evolving fraud patterns over simulated time periods, allowing models to be tested for adaptability rather than static performance. Research into online learning approaches, such as continual learning frameworks or drift detection mechanisms, could empower fraud detection models to adjust dynamically to emerging threats without requiring complete retraining.

Fourth, development of adaptive, lightweight hybrid architectures is recommended to meet operational constraints in real-time banking environments. While the hybrid LSTM-XGBoost model demonstrated promising accuracy and interpretability, optimizing model complexity remains essential for practical deployment. Techniques such as model distillation, pruning, and quantization could help produce streamlined versions of hybrid models without significant loss of detection capability, thereby expanding the accessibility of advanced fraud detection even to institutions with limited computational infrastructure.

Finally, comparative evaluations across multiple institutions and regional contexts are necessary to validate the robustness and generalizability of anomaly detection systems. Studies incorporating multi-bank data, varied regulatory environments, and demographic diversity would offer richer insights into how models must be tailored or adapted to different operational landscapes. Such cross-context investigations could also shed light on region-specific fraud typologies, informing the development of localized detection strategies.

CONCLUSION

In the age of digital finance, online banking has become the dominant medium for monetary transactions, reshaping the way individuals interact with financial systems and increasing the complexity of fraud detection. This thesis originated from the urgent need to address behavioral anomalies in such banking environments, especially in contexts where access to real transaction data is restricted and anomaly classes are highly imbalanced. Drawing on the synergy between expert-driven data simulation and advanced ML techniques, the study proposed a comprehensive hybrid approach that enhances the accuracy, reliability, and interpretability of anomaly detection in online banking.

By building upon and extending both domestic and international research, this study contributes a novel theoretical and practical framework for anomaly detection. It introduces a synthetic data generation engine that mimics user-level transaction behaviors and simulates fraudulent activities based on expert-defined rules. This method addresses a key gap in existing literature, where few studies explicitly demonstrate how behavioral anomaly data can be systematically constructed in the absence of real-world labels. Simultaneously, the research develops and compares three detection models: traditional ML, DL with LSTM, and a hybrid model across a rigorously engineered dataset that reflects the multifaceted nature of banking transactions.

The research successfully achieved several core objectives: (i) It systematized foundational knowledge in anomaly detection, drawing from financial, ML, and behavioral modeling literature, and introduced a hybrid model that merges sequential LSTM embeddings with tabular features using XGBoost classifiers; (ii) It provided a synthetic data set that mimic the real behavior of normal users in online banking transactions; (iii) It implemented and validated the performance of the proposed models using a large synthetic dataset and advanced evaluation metrics, revealing the hybrid model as the most effective solution in terms of F1-score and ROC-AUC, particularly under imbalanced data conditions. (iv) It generated research implications for both academic and applied domains, suggesting how synthetic data, hybrid modeling can support scalable and trustworthy fraud detection systems in financial institutions.

The empirical findings highlight that the hybrid model outperforms both ML and DL models when evaluated on both sensitivity and specificity. While LSTM excels at capturing behavioral shifts, and ML models are interpretable and efficient, it is the combination of both that yields a balanced and powerful detection mechanism. Moreover, the synthetic data engine not only supports training but also functions as a platform for future testing and experimentation, demonstrating its dual utility in model development and validation.

In terms of theoretical value, this thesis advances the field by proposing a functional integration of time-series DL with traditional tabular classifiers. It challenges the siloed application of single-class models and offers an approach that embraces the complex temporal nature of fraud without sacrificing clarity in model interpretation. The design of the synthetic dataset and the transparent architecture of the hybrid model make this framework replicable and adaptable to other sectors where real-time fraud detection is critical.

From a practical standpoint, the research equips financial institutions, analysts, and system developers with an operational roadmap for deploying hybrid fraud detection systems. The model's interpretability aligns with international compliance requirements, and its adaptability allows it to evolve alongside new fraud patterns and transaction trends. Additionally, the modular design of the synthetic data generator enables iterative improvements and customized testing for varied organizational contexts.

Despite its promising results, this study acknowledges several limitations. The synthetic dataset, while robust and behaviorally diverse, may not fully replicate the nuance of actual customer psychology or external triggers such as policy changes or economic shocks. The model also focuses primarily on transaction-based anomalies, excluding account-level or biometric patterns that could further enhance fraud detection accuracy. These limitations open new directions for future research.

In summary, this thesis offers both theoretical and practical contributions to the field of financial fraud detection. It provides a validated hybrid model grounded in behavioral insight and technological integration, along with a replicable data simulation framework that addresses core challenges in data scarcity and class imbalance. Though further refinements are necessary, especially through future access to real-world data or by expanding to multi-modal detection systems, the research establishes a solid foundation for the ongoing evolution of secure, explainable, and intelligent fraud detection in online banking.