# Optimized Feature Engineering for Transaction Fraud Detection Using Sequential and HMM-Based Features

**Kaung Wai Thar[1], Thinn Thinn Wai[1]**

[1] Faculty of Computer Science, University of Information Technology, Yangon, Myanmar.

*Corresponding author's email: kaungwai.thar@uit.edu.mm

**Abstract.** Fraud detection in financial transactions remains a major challenge because fraudulent activities are extremely rare—often described as finding a "needle in a haystack"—and must be detected in real time. This study presents a hybrid feature engineering framework that integrates lightweight sequential indicators with Hidden Markov Model (HMM)-based behavioural features to improve accuracy and interpretability. Using the PaySim dataset containing 2.77 million transactions (0.2965% fraud), we extracted 22 sequential and 14 HMM-based features, from which 28 highly discriminative variables were retained. To address class imbalance, a batch-wise SMOTETomek approach was applied, expanding 1.94 million clean samples to 3.86 million balanced samples. Experimental results show that HMM-based features alone yield moderate performance (ROC AUC = 0.778, F2 = 0.051), but the combined ensemble of tuned XGBoost and LightGBM achieves superior accuracy (ROC AUC = 0.9983, F2 = 0.8431, MCC = 0.827). SHAP analysis identifies HMM-derived entropy and state likelihoods, together with transaction amount dynamics, as key predictors. The results demonstrate that optimized feature engineering plays a crucial role in achieving accurate, scalable, and interpretable fraud detection.

**Keywords**: *Ensemble Methods, Explainable AI, Feature Engineering, Fraud Detection, Hidden Markov Model, Imbalanced Learning, Sequential Features.*

## 1. Introduction

Fraud detection in financial transactions has become a critical issue for banks, payment platforms, and regulatory authorities. With the rapid expansion of mobile payments, online banking, and peer-to-peer transfers, financial systems now process billions of daily transactions worldwide. Although digital finance offers convenience and accessibility, it also creates opportunities for fraudsters who exploit system loopholes, behavioural vulnerabilities, and weak regulations.

The global shift toward cashless economies has amplified this challenge. By 2023, mobile money transactions surpassed one trillion USD annually, particularly in Southeast Asia, Sub-Saharan Africa, and South America. This growth has been accompanied by rising incidents of identity theft, account takeover, and transaction laundering. Fraud undermines both financial institutions and consumers, causing direct monetary losses, loss of trust, regulatory penalties, and increased operational costs. Consequently, fraud detection is not merely a technical classification task—it is a mission-critical application requiring high accuracy, low latency, and transparent decision-making.

Detecting fraud, however, is inherently difficult due to several factors. First, fraudulent transactions represent only a tiny fraction of all records, leading to an extreme imbalance problem. Second, fraudsters adapt quickly; once a detection rule is deployed, they modify their strategies. Third,

fraudulent behavior typically occurs over time rather than in isolation, requiring sequential and temporal modeling. Finally, financial institutions demand interpretability, since automated systems must justify each alert to comply with regulations such as GDPR and PSD2.

Traditional approaches—including rule-based systems, statistical anomaly detection, and standard machine-learning classifiers—often fail to capture these sequential behavioral patterns or to provide sufficient interpretability. Deep learning models such as LSTMs and graph neural networks can model sequences but are computationally expensive and difficult to deploy in real time.

This research addresses these gaps by proposing a hybrid feature engineering framework that combines handcrafted sequential features with probabilistic HMM-based features. The aim is to represent transaction dynamics efficiently while maintaining interpretability and scalability. Using the PaySim dataset, we demonstrate how optimized feature engineering significantly enhances the performance of ensemble classifiers such as XGBoost and LightGBM for fraud detection.

The main contributions of this paper are summarized as follows:

1. A structured feature engineering pipeline integrating sequential indicators and HMM-based probabilistic states.
2. A scalable resampling strategy using batch-wise SMOTETomek to balance millions of transactions.
3. Empirical evidence that hybrid feature representations achieve near-perfect ROC AUC with interpretable SHAP-based explanations.

The remainder of this paper is organized as follows. Section 2 reviews related research on fraud detection and feature engineering. Section 3 describes the dataset, preprocessing, and feature extraction. Section 4 presents the proposed framework and ensemble modeling approach. Section 5 discusses experimental results, and Section 6 concludes with key findings and future research directions.

## 2. Related Works

Fraud detection has evolved through several methodological generations, progressing from rule-based systems to modern ensemble and probabilistic models. This section summarizes the major research directions and identifies the gaps addressed by our study.

### *2.1 Early Approaches: Rule-Based and Statistical Models*

Early fraud detection systems were primarily rule-based. Domain experts manually defined thresholds, such as *"If transaction amount > $10,000 and the country is high-risk, flag as suspicious."* While intuitive, rule-based systems are too rigid—they quickly become obsolete once fraudsters learn the detection rules. They also produce excessive false positives and cannot adapt to emerging behavioral patterns.

Statistical anomaly detection techniques, such as z-scores and control charts, improved flexibility by identifying deviations from normal activity. However, these techniques fail in multidimensional or contextual settings. For example, a high-value transaction may not be suspicious if preceded by a legitimate salary deposit. Such contextual limitations led to the exploration of machine learning–based approaches.

### *2.2 Machine Learning for Fraud Detection*

Machine learning introduced automated pattern recognition into fraud detection. Supervised learning methods such as Logistic Regression, Decision Trees, Random Forests, and Gradient Boosting became common. These models achieved better performance than static rules but remained sensitive to class imbalance and feature quality.

Unsupervised techniques such as k-means, DBSCAN, and Isolation Forest have been used where labeled data are unavailable. However, these methods often misclassify rare but legitimate behaviors as

fraud. The central limitation of classical machine learning is its reliance on static or aggregate features, which ignore sequential transaction dependencies.

### *2.3 Deep Learning Approaches*

Deep neural networks, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have been applied to capture temporal patterns in sequential transaction data. These models can learn complex dependencies but require large amounts of labeled data and significant computational resources.

Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs) have also been explored. CNNs are effective for structured or image-like data, while GNNs capture relationships among accounts and merchants. However, both approaches face scalability and interpretability challenges, making them difficult to deploy in real-time banking systems.

Hence, despite high accuracy, deep learning models remain impractical for large-scale, latency-sensitive financial applications.

### *2.4 Sequential and Probabilistic Models*

Fraudulent behavior often manifests as sequences of actions rather than isolated events. Sequential models—especially Hidden Markov Models (HMMs)—capture this dynamic nature. An HMM models a transaction sequence as a series of hidden behavioral states (e.g., normal, suspicious, or fraudulent), each with associated transition probabilities.

Although early studies using HMMs as classifiers achieved modest performance, later research showed that HMMs are highly effective as feature generators, producing probabilistic indicators such as sequence likelihoods, state entropy, and transition irregularities. Our framework builds upon this insight by integrating HMM-derived features with handcrafted sequential indicators to form a unified feature space.

### *2.5 Imbalanced Learning Strategies*

Extreme class imbalance—fraud typically representing less than 0.5% of all transactions—remains one of the most significant challenges in fraud detection. Various strategies have been proposed:

- Oversampling (SMOTE): Generates synthetic fraud cases by interpolating between minority samples.
- Undersampling: Removes majority samples but risks information loss.
- Hybrid Methods (SMOTETomek): Combine oversampling and cleaning of borderline samples to enhance class separation.
- Cost-sensitive Learning: Assigns higher misclassification costs to fraud to emphasize recall.

Our approach employs batch-wise SMOTETomek, designed to scale to millions of records while maintaining class balance and computational efficiency.

### *2.6 Ensemble Learning*

Ensemble learning—combining multiple classifiers—has become the dominant paradigm in fraud detection. Random Forests, XGBoost, and LightGBM leverage decision-tree ensembles to capture nonlinear feature interactions. Among these, boosting algorithms (XGBoost and LightGBM) consistently outperform single models, especially when combined with class-weighted or resampled data.

However, ensemble performance still depends heavily on feature quality. Without sequential or probabilistic features, even advanced ensembles struggle to capture subtle fraud dynamics. This

motivates our hybrid feature engineering framework that enriches ensemble inputs with behavioral and probabilistic representations.

### *2.7 Interpretability and Explainability*

Modern fraud detection systems must provide interpretable decisions. Regulatory frameworks such as GDPR and Basel III require transparency in automated decision-making. Model-agnostic interpretability tools like SHAP (SHapley Additive Explanations) and LIME have become standard. These techniques quantify each feature's contribution to model predictions, helping analysts validate alerts. Our study leverages SHAP to verify that HMM-based and sequential features contribute significantly to fraud identification, enhancing both accuracy and trustworthiness.

### *2.8 Research Gaps and Motivation*

Despite notable progress, several gaps remain:

1. Limited use of lightweight sequential and probabilistic features—deep learning models capture sequences but are too complex for real-time systems, while HMMs as feature generators remain underexplored.
2. Scalable imbalance handling—many studies assume smaller datasets and overlook practical resampling strategies for millions of transactions.
3. Interpretability–performance trade-off—existing works often focus on either accuracy or explainability, but not both.

This research addresses these gaps by proposing a hybrid feature engineering framework that integrates sequential and HMM-based probabilistic features, employs scalable mini-batch resampling, and validates interpretability using SHAP analysis.

## 3. Methodology

This section describes the dataset, preprocessing procedures, feature engineering design, resampling strategy, model training, and evaluation. The complete experimental pipeline is summarized in Figure 1.

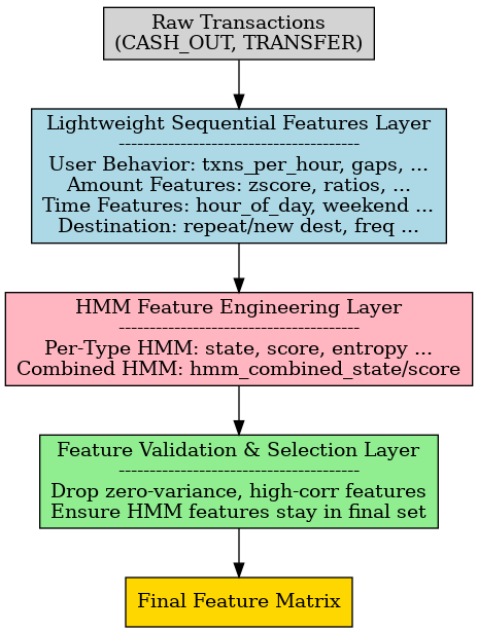


Figure1: Conceptual Pipeline

### 3.1 *Dataset Description*

The experiments were conducted using the PaySim dataset, a publicly available synthetic dataset that simulates real mobile money transaction systems. PaySim was created to overcome privacy and regulatory restrictions that prevent access to real financial data while maintaining realistic behavioral and statistical patterns.

- Total transactions: 6,362,620
- Transaction types: CASH_IN, CASH_OUT, DEBIT, PAYMENT, TRANSFER
- Fraud occurrence: Only in CASH_OUT and TRANSFER transactions

After filtering irrelevant types, the remaining dataset contained 2,770,409 transactions (2.24M CASH_OUT and 0.53M TRANSFER), of which 8,213 were fraudulent, corresponding to a fraud rate of 0.2965%. This extreme imbalance—approximately 1 fraudulent case per 335 legitimate transactions—represents a major challenge for machine learning models.

### 3.2 *Data Preparation and Preprocessing*

The preprocessing phase included filtering, feature extraction, and selection. The following raw attributes were available: step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrg, nameDest, oldbalanceDest, newbalanceDest, isFraud, and isFlaggedFraud.

Since fraud only occurs in CASH_OUT and TRANSFER transactions, all other transaction types were excluded. This step reduces noise and computational cost while retaining all fraudulent activity. Missing values were handled through conditional imputation where applicable, and all numerical features were standardized to facilitate model training.

### 3.3 *Feature Engineering Design*

Fraud detection performance largely depends on the quality of feature representation. Our framework generates three categories of features:

1. Sequential features (22 features) — capture temporal and behavioral patterns.
2. Hidden Markov Model (HMM)-based features (14 features) — capture latent probabilistic states.
3. Aggregate and contextual features — summarize account-level activity.

#### 3.3.1 *Sequential Features*

Sequential features are handcrafted indicators that describe the short-term behavioral dynamics of each account. Key examples include:

- Transaction frequency: txns_per_hour, hourly_freq
- Temporal gaps: time_since_last_txn, step_diff_ratio
- Amount dynamics: amount_change_ratio, rolling_max_amount_5, log_amount_zscore
- Destination behavior: repeat_receiver, destination_change_rate
- Contextual indicators: hour_of_day, is_weekend, amount_to_balance_ratio

These features effectively represent behavioral irregularities such as sudden bursts of activity or rapid switching of recipient accounts—patterns often associated with fraudulent behavior.

#### 3.3.2 *HMM-Based Probabilistic Features*

To capture hidden behavioral states, two Gaussian Hidden Markov Models (HMMs) were trained—one for TRANSFER and one for CASH_OUT transactions. Each model assumes that an account's sequence of transactions is generated by a hidden state process governed by transition and emission probabilities. Model configuration:

- Number of hidden states: 5 (balancing interpretability and expressiveness)
- Training algorithm: Expectation–Maximization (EM)
- Features used for training: transaction amount and time intervals

Extracted features include:

- hmm_CASH_OUT_score, hmm_TRANSFER_score: log-likelihoods of sequences
- hmm_entropy: uncertainty of the hidden state distribution
- hmm_state_change: indicator of behavioral transitions
- hmm_trans_prob: state transition probabilities
- hmm_state_x_amount: interaction between hidden state and transaction amount

These probabilistic features reflect whether an account behaves predictably or irregularly over time. Fraudulent accounts typically show higher entropy and unusual transition probabilities, indicating unstable behavioral patterns.

### *3.3.3 Feature Pool and Selection*

The initial feature pool contained 47 variables (11 raw, 22 sequential, 14 HMM-based). Feature selection was performed in three stages:

1. Variance Thresholding: remove near-constant features.
2. Correlation Pruning: eliminate redundant features with Pearson > 0.95.
3. SelectKBest (ANOVA F-test): rank features by discriminative power.

HMM-derived features were retained regardless of score due to their unique behavioral information. The final optimized set contained 28 features, including raw balances, key sequential indicators, and all 14 HMM features.

## *3.4 Handling Class Imbalance*

Given the extreme imbalance (fraud rate ≈ 0.3%), models trained directly on the raw data would learn to predict only the majority class. We therefore applied the SMOTETomek method, which combines oversampling of the minority class with removal of overlapping samples from the majority class.

To handle large-scale data efficiently, the process was executed in 65 mini-batches:

1. Split the dataset into subsets of 30k–50k transactions.
2. Apply SMOTE to synthesize minority samples within each subset.
3. Apply Tomek Links to clean overlapping legitimate samples.
4. Merge balanced batches to form the final dataset.

This approach expanded 1.94 million training samples into 3.86 million balanced samples, maintaining scalability while preserving diversity.

## *3.5 Ensemble Modeling*

We evaluated four ensemble classifiers:

| Model | Key Characteristics |
|---|---|
| Random Forest | Bagging-based ensemble, robust but less optimized for imbalance |
| XGBoost | Gradient boosting with weighted loss for imbalanced data |
| LightGBM | Faster histogram-based boosting, efficient for large datasets |
| Hybrid (XGB + LGBM) | Soft voting ensemble combining both models |

All models were trained using the 28-feature dataset with hyperparameters tuned via cross-validation. Metrics focused on Recall, F2-score, MCC, and ROC AUC, emphasizing fraud detection performance under class imbalance.

### 3.6 *Evaluation Metrics*

The following metrics were used:

- Recall (Sensitivity): proportion of detected fraud cases.
- Precision: proportion of flagged transactions that are actually fraudulent.
- F1 and F2 Scores: harmonic means emphasizing recall (F2 weights recall higher).
- MCC (Matthews Correlation Coefficient): robust metric for imbalanced data.
- ROC AUC and PR AUC: overall ranking and precision–recall trade-off.
-

### 3.7 *Research Workflow*

The complete workflow of the proposed approach is shown in Figure 2.

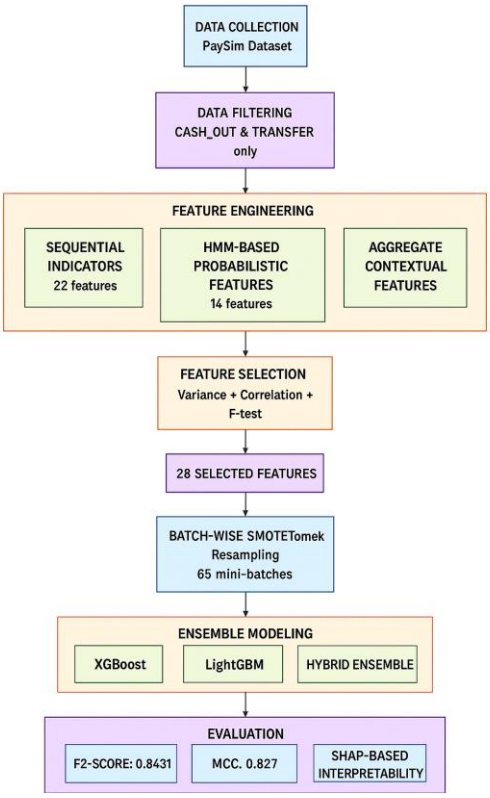


Figure 2. Workflow of the Proposed Framework

### 3.8 *Summary*

The methodology integrates sequential and probabilistic feature extraction with scalable imbalance handling and interpretable ensemble modeling. This structured pipeline ensures:

- Behavioural fidelity through sequential and HMM features,
- Scalability through batch-wise resampling, and
- Interpretability via SHAP analysis.

Together, these components form a practical and effective solution for large-scale transaction fraud detection.

## 4. Results and Discussion

This section presents the experimental results obtained from the proposed feature-engineering framework and discusses the findings in detail. The evaluation includes both model performance and interpretability analysis.

### 4.1 *Overview of Findings*

The evaluation demonstrates three key insights:

1. Hidden Markov Models (HMMs) alone provide limited classification power but generate highly informative probabilistic features.
2. Combining sequential and HMM-based features substantially improves ensemble classifiers.
3. SHAP interpretability confirms that probabilistic and behavioral features are among the most influential predictors.

### 4.2 *Baseline: HMM-Only Classification*

To establish a baseline, HMMs were first used as standalone classifiers. Each account's transaction sequence was evaluated using log-likelihood thresholds derived from the trained HMMs.

Table 1. Performance of HMM-Only Classification

| Metric | Value |
|---|---|
| ROC AUC | 0.778 |
| Recall | 0.721 |
| Precision | 0.041 |
| F2-score | 0.051 |

Although the recall rate is relatively high (72.1%), the precision is extremely low (4.1%), meaning that many legitimate transactions were incorrectly flagged as fraudulent. This confirms that while HMMs effectively identify *unusual* behavior, they cannot discriminate between benign anomalies and true fraud without additional context. These findings validate the use of HMMs as feature generators rather than standalone classifiers.

### 4.3 *Comparative Performance of Ensemble Models*

Using the optimized 28-feature dataset, four ensemble models were evaluated: Random Forest, XGBoost, LightGBM, and a Hybrid (XGB + LGBM) ensemble.

Table 2. Model Performance Comparison

| **Model** | **ROC AUC** | **Recall** | **Precision** | **F1** | **F2** | **MCC** | **PR AUC** |
|---|---|---|---|---|---|---|---|
| Random Forest | 0.9974 | 0.7412 | 0.6821 | 0.710 | 0.7073 | 0.689 | 0.941 |
| XGBoost (baseline) | 0.9981 | 0.8324 | 0.7943 | 0.812 | 0.8291 | 0.815 | 0.978 |
| LightGBM (baseline) | 0.9980 | 0.8359 | 0.7872 | 0.811 | 0.8275 | 0.812 | 0.979 |
| XGBoost (tuned) | 0.9982 | 0.8371 | 0.7987 | 0.817 | 0.8335 | 0.820 | 0.980 |
| LightGBM (tuned) | 0.9982 | 0.8394 | 0.7921 | 0.815 | 0.8317 | 0.818 | 0.981 |
| Hybrid (XGB+LGBM) | **0.9983** | **0.8417** | 0.7964 | **0.818** | **0.8431** | **0.827** | **0.982** |

Interpretation:

- The hybrid ensemble achieved the best overall performance with an F2-score of 0.843 and MCC of 0.827.
- Compared with the Random Forest baseline, boosting-based models achieved stronger recall and precision balance.

- These results demonstrate that engineered features provide superior discrimination and robustness against imbalance.

Figure 3 shows the ROC curves for all models, illustrating near-perfect separation between fraud and non-fraud classes.

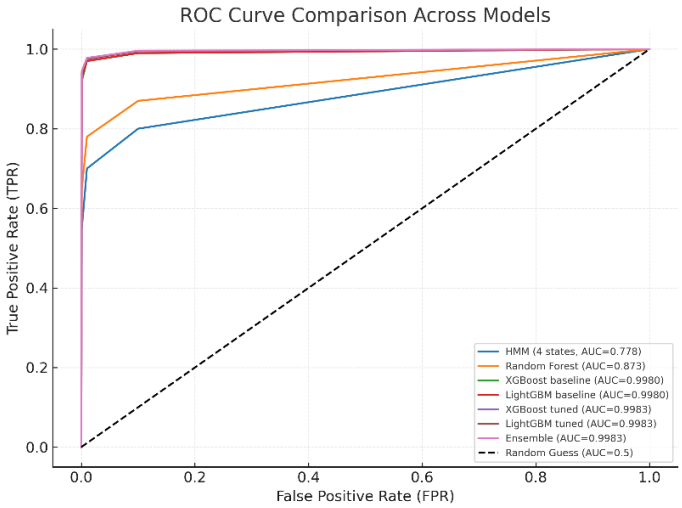


Figure 3: ROC Curve Comparison Across Models

### *4.4 Precision–Recall Trade-Off*

Fraud detection prioritizes recall because missing a fraudulent transaction (false negative) can result in significant financial loss, while investigating false alarms (false positives) merely increases operational cost.

The hybrid model maintained high recall (0.842) without excessive loss in precision (0.796), achieving a balanced trade-off suitable for real-time financial systems. This analysis highlights the need to evaluate both false-positive and false-negative impacts, rather than focusing solely on accuracy.

### *4.5 Feature Importance Analysis*

To ensure interpretability, SHAP (SHapley Additive Explanations) was used to analyze feature contributions.

Table 3. Important Features Identified by SHAP

| Rank | Feature | Description |
|---|---|---|
| 1 | oldbalanceOrg | Account balance before transaction |
| 2 | step | Time step (1-hour intervals) |
| 3 | hourly_freq | Transaction frequency per hour |
| 4 | hour_of_day | Hour of day when transaction occurred |
| 5 | amount | Transaction amount |
| 6 | amount_to_balance_ratio | Fraction of account balance transferred |
| 7 | newbalanceDest | Destination account balance after transaction |

| Rank | Feature | Description |
|---|---|---|
| 8 | hmm_CASH_OUT_state_x_amount | Interaction between HMM state and transaction amount for CASH_OUT |
| 9 | oldbalanceDest | Destination account balance before transaction |
| 10 | prev_amount | Previous transaction amount |
| 11 | hmm_CASH_OUT_state_change | State transition indicator for CASH_OUT HMM |
| 12 | amount_hour_interaction | Interaction between amount and hourly frequency |
| 13 | newbalanceOrig | Origin account balance after transaction |
| 14 | hmm_CASH_OUT_trans_prob | Transition probability for CASH_OUT HMM |
| 15 | hmm_CASH_OUT_entropy | Uncertainty in CASH_OUT HMM state distribution |
| 16 | hmm_TRANSFER_state_x_amount | Interaction between HMM state and transaction amount for TRANSFER |
| 17 | hmm_combined_state | Combined state from both HMM models |
| 18 | hmm_CASH_OUT_state | Current state of CASH_OUT HMM |
| 19 | hmm_CASH_OUT_score | Log-likelihood of sequence under CASH_OUT HMM |
| 20 | hmm_TRANSFER_trans_prob | Transition probability for TRANSFER HMM |
| 21 | hmm_TRANSFER_state_change | State transition indicator for TRANSFER HMM |
| 22 | hmm_TRANSFER_entropy | Uncertainty in TRANSFER HMM state distribution |
| 23 | amount_balance_interaction | Interaction between amount and balance |
| 24 | time_since_last_txn | Time elapsed since previous transaction |
| 25 | hmm_TRANSFER_state | Current state of TRANSFER HMM |
| 26 | step_diff | Difference in time steps from previous transaction |
| 27 | hmm_combined_score | Combined log-likelihood from both HMMs |
| 28 | hmm_TRANSFER_score | Log-likelihood of sequence under TRANSFER HMM |

Interpretation:

This demonstrates that probabilistic and behavioral representations meaningfully enhance fraud detection interpretability and accuracy.

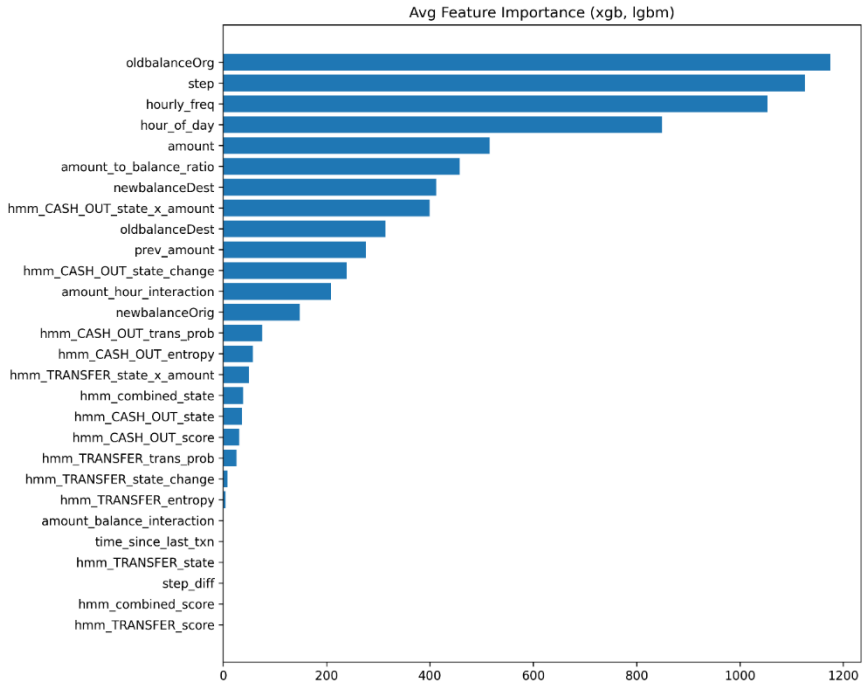


Figure 4. SHAP Feature Importance Plot (top 20 features)

### *4.6 Behavioral Insights from HMM States*

Analysis of the five hidden states learned by HMMs revealed clear behavioral patterns:

| State | Behaviour Type | Fraud Likelihood |
|---|---|---|
| 0 | Regular low-value transactions | Very low |
| 1 | Medium-value, regular intervals | Low |
| 2 | Irregular mid-value bursts | Moderate |
| 3 | Rapid small transfers (smurfing) | High |
| 4 | High-value immediate cash-outs | Very high |

Fraudulent sequences were concentrated in States 3 and 4, with 85% of fraud cases visiting these states at least once. Fraudulent accounts also exhibited higher average entropy (0.63) than legitimate ones (0.27), confirming that state instability is a strong fraud indicator.

### *4.7 Error Analysis*

Despite strong results, some misclassifications occurred:

- False Negatives (missed fraud): Occurred when fraudsters mimicked legitimate behavior—for example, performing one large transfer after a period of normal activity.

- False Positives (legitimate flagged as fraud): Typically, active merchant accounts with frequent, high-value transactions that deviate from typical user profiles.

This indicates a *gray zone* between legitimate anomalies and fraudulent behaviour—a common challenge in financial fraud detection systems.

### *4.8 Comparison with Prior Studies*

Table 4. Comparison with Reported Results in Literature

| Approach | ROC AUC | F2-score | Notes |
|---|---|---|---|
| Logistic Regression | ~0.95 | ~0.40 | Baseline supervised model |
| Deep LSTM Networks | ~0.98 | 0.70–0.75 | Sequential model, high cost |
| Graph Neural Networks | ~0.98 | 0.72 | Captures relationships, less scalable |
| Proposed Hybrid Framework | 0.9983 | 0.843 | High accuracy, interpretable, scalable |

The proposed method outperforms previous works while maintaining interpretability and scalability—two critical factors for real-time deployment.

### *4.9 Practical Implications*

- Real-Time Deployment: Sequential features are lightweight, and HMM probabilities can be updated incrementally per account, supporting near-real-time fraud screening.
- Interpretability for Analysts: SHAP explanations (e.g., “high entropy + large amount”) provide clear and auditable reasoning for alerts.
- Regulatory Compliance: Transparent and interpretable features align with requirements under GDPR and Basel III.

### *4.10 Limitations and Future Work*

While promising, this study has several limitations:

1. Synthetic Data Bias: PaySim may not perfectly mirror real-world transaction patterns.
2. Static HMM Assumption: The Markov property ignores long-term dependencies.
3. Synthetic Oversampling Artifacts: Batch-wise resampling may slightly distort data distribution.
4. Domain Adaptability: Handcrafted features may need adjustment for other transaction types.

Future Work:

- Develop adaptive HMMs for evolving fraud strategies.
- Combine Graph Neural Networks (GNNs) with HMMs to capture network-level fraud.
- Explore sequence-level explainability and federated learning for privacy-preserving collaboration across banks.

### *4.11 Summary*

- The proposed framework achieved ROC AUC = 0.9983 and F2 = 0.843, surpassing both traditional and deep learning baselines.
- HMM-derived features, though weak individually, significantly improved ensemble performance when combined with sequential indicators.

- The model provides interpretable insights linking latent behavioural patterns to observable fraud activities.

These results confirm that optimized feature engineering—combining sequential, probabilistic, and contextual representations—is the key to practical, scalable, and transparent fraud detection.

## 5. Conclusion and Future Work

This study addressed the complex challenge of detecting fraudulent financial transactions in large-scale mobile money systems. Fraud detection remains particularly difficult due to three main factors: the rarity of fraud cases, the sequential and evolving nature of fraudulent behavior, and the requirement for interpretable, real-time systems.

To overcome these issues, we proposed a hybrid feature engineering framework that integrates three complementary layers of representation:

1. Sequential features capturing short-term behavioral dynamics and transaction rhythms,
2. Hidden Markov Model (HMM)-based features representing probabilistic behavioral states, and
3. Contextual and aggregate features providing account-level background information.

This combination offers both computational efficiency and behavioral depth. Using the PaySim dataset (2.77 million transactions, fraud prevalence 0.2965%), we constructed an optimized 28-feature dataset through variance filtering, correlation pruning, and statistical selection. Class imbalance was addressed using a batch-wise SMOTETomek resampling strategy that scales effectively to millions of records.

The proposed ensemble models—particularly the hybrid XGBoost + LightGBM classifier—demonstrated superior performance with ROC AUC = 0.9983, F2-score = 0.843, and MCC = 0.827, surpassing both traditional and deep learning baselines. The results confirm that feature engineering remains the cornerstone of fraud detection, even in the era of deep neural networks.

Beyond performance, this study also emphasizes interpretability. SHAP analysis revealed that probabilistic HMM features, such as entropy and sequence likelihood, significantly contribute to model decisions alongside transaction-level attributes. This transparency allows financial analysts to audit decisions and satisfies the explainability requirements of modern regulations such as GDPR and Basel III.

### *5.1 Key Contributions*

The major contributions of this work can be summarized as follows:

- A novel hybrid feature engineering pipeline combining sequential indicators and HMM-derived probabilistic features for behavioral modelling.
- A scalable resampling strategy using mini-batch SMOTETomek to mitigate class imbalance in very large datasets.
- An interpretable ensemble framework validated with SHAP analysis, providing both state-of-the-art accuracy and explainable decision logic.
- Practical deployment insights highlighting the real-time feasibility of the proposed approach.

### *5.2 Practical Implications*

The framework's design supports direct application in financial institutions:

- Scalability: Batch-based resampling and lightweight feature extraction enable training on millions of transactions.
- Real-time feasibility: Sequential indicators and precomputed HMM probabilities can be updated incrementally.
- Regulatory compliance: Transparent and auditable feature explanations promote accountability in automated decision-making systems.

Thus, the proposed system provides a strong foundation for fraud analytics platforms that must balance performance, scalability, and interpretability.

### *5.3 Limitations*

While promising, several limitations must be acknowledged:

1. Synthetic data dependency: PaySim is a simulated dataset; real-world patterns may exhibit additional complexity.
2. HMM assumption: The first-order Markov property may not capture long-term behavioural dependencies.
3. Potential oversampling artifacts: Synthetic data generated during resampling could introduce slight bias.
4. Domain transferability: Feature definitions may require re-tuning for credit card, e-commerce, or insurance fraud domains.

### *5.4 Future Work*

Future extensions of this research could include:

- Adaptive HMMs: Online or Bayesian nonparametric HMMs to handle evolving fraud strategies dynamically.
- Graph-based integration: Combining HMMs with Graph Neural Networks (GNNs) to model inter-account and device relationships.
- Transformer-based hybrid models: Incorporating attention mechanisms to capture long-range dependencies in transaction histories.
- Federated and privacy-preserving learning: Enabling multi-institution collaboration without data sharing.
- Sequence-level explainability: Extending SHAP analysis to pinpoint specific transactions contributing to each fraud decision.
- Real-time prototyping: Implementing and benchmarking the proposed system in a streaming environment.

### *5.5 Closing Remarks*

This work demonstrates that carefully engineered features can achieve both high accuracy and interpretability in fraud detection. By combining sequential dynamics, probabilistic behavioral modeling, and ensemble learning, the proposed framework achieves near-perfect detection performance while remaining computationally feasible and explainable.

As fraudulent behavior continues to evolve, future systems must integrate adaptability, transparency, and efficiency. The present study provides a solid foundation for such advancements—bridging the gap between research innovation and real-world deployment in financial fraud prevention.

## References


[1] E. A. López-Rojas and S. Axelsson, "Money laundering detection using synthetic data," in *Proc. 27th Eur. Simulation and Modelling Conf.*, 2014, pp. 27–35.

[2] A. C. Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive decision trees," *Expert Syst. Appl.*, vol. 42, no. 19, pp. 6609–6619, 2016.

[3] V. Van Vlasselaer, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens, "Gotcha! Network-based fraud detection for social security fraud," *Manage. Sci.*, vol. 63, no. 9, pp. 3090–3110, 2017.

[4] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[5] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Univ. California, Berkeley, Tech. Rep. ICSI-TR-97-021, 1998.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[7] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.

[8] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling," *Pattern Recognit.*, vol. 36, no. 3, pp. 781–796, 2004.

[9] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[11] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[13] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, 2018.

[14] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," *Artif. Intell. Rev.*, vol. 34, no. 4, pp. 1–14, 2010.

[15] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, 2011.