www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 1124-1130

e-ISSN : 2583-1062
Impact Factor : 7.001

MACHINE LEARNING BASED ENHANCEMENT OF REAL-TIME FRAUD DETECTION IN E-PAYMENT SYSTEMS USING HIDDEN MARKOV MODEL(HMM)

V. Sumedh[1], N. Sumith[2], K. Surya Kiran[3], M. Suryam[4], R. Swathi[5], Prof. P. Bhavani[6]

[1,2,3,4,5]B. Tech School of Engineering MallaReddy University Hyderabad, India.

[6]Professor School of Engineering Malla Reddy University Hyderabad, India.

2111CS020577@mallaredd yuniversity.ac.in, 2111CS020578@mallareddyuniversity.ac.in, 2111CS020579@malla reddyuniversity.ac.in 2111CS020580@mallareddyuniversity.ac.in yuniversity.ac.in, 2111CS020581@mallaredd yuniversity.ac.in, p.bhavani@mallareddyuniversity.ac.in

DOI: https://www.doi.org/10.58257/IJPREMS39308

ABSTRACT

The rapid growth of online transactions has increased the risk of fraudulent activities, posing significant challenges to businesses and consumers. Detecting and preventing online payment fraud is essential to maintain the security and trustworthiness of digital payment systems. This project aims to develop a machine learning-based system capable of identifying fraudulent transactions in real-time with high accuracy. The solution utilizes both supervised and unsupervised learning methods to analyze transaction data and detect irregular patterns indicative of fraud. Machine learning algorithms such as Random-Forest, Gradient Boosting, and Neural Networks are employed to classify transactions as legitimate or suspicious. Techniques like feature extraction, anomaly detection, and ensemble modeling are applied to improve detection performance. Additionally, data pre-processing steps, including balancing techniques like SMOTE, help manage class imbalances in transaction datasets.

1. INTRODUCTION

Imagine shopping online or sending money through a mobile app—e-payment systems have made these transactions quick and easy, changing the way we handle money in today's digital world. But with this convenience comes a big challenge: fraud. Scammers are getting smarter, using tricks like stealing identities or making unauthorized payments, and it's costing us a lot—over $40 billion worldwide in 2023 alone! Worse, these frauds make people hesitant to trust online payments. The old-school methods to catch fraud, like setting rigid rules to flag suspicious transactions, just aren't keeping up. They often flag innocent purchases by mistake or miss the sneaky ones, and they're too slow to stop fraud in its tracks. That's where our project, "Machine Learning-Based Enhancement of Real-Time Fraud Detection in E-Payment Systems Using Hidden Markov Model (HMM)," comes in. We're building a smarter system that uses a machine learning tool called Hidden Markov Model (HMM) to spot fraud instantly by looking at patterns in your transactions—like how often you pay, where, and how much. It's like having a super-fast detective that learns your habits and flags anything fishy right away. Our goal is to make e-payments safer, more reliable, and trustworthy, so you can shop or send money without worry, knowing fraudsters won't get away with it.

The old-school ways of catching fraud just aren't keeping up. Most systems use rules to spot trouble—like flagging a purchase over $500 or one from a new country. But these rules are too stiff. They often flag your legit purchases, like that new phone you bought as a gift, while missing the sneaky frauds that look normal at first glance, like a string of small charges that add up fast. On top of that, they're slow. By the time they catch something fishy, the money's often already gone, leaving you and the banks to deal with the mess. These outdated methods also need constant updates by experts to stay relevant, which is expensive and can't keep pace with how fast fraudsters change their tricks. It's clear we need a smarter, faster way to protect our money in the digital world.

2. LITERATURE REVIEW

[1] " Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review"(2022): This systematic literature review synthesizes 93 studies from 2010 to 2022, exploring machine learning techniques for financial fraud detection, including e-payment systems. The authors categorize methods into supervised, unsupervised, and hybrid approaches, with Hidden Markov Models (HMMs) highlighted for their ability to model sequential transaction data and user behavior.

" Credit Card Fraud Detection Using Machine Learning: A Survey" (2020): Lucas et al. survey ML techniques for credit card fraud detection, focusing on challenges like dataset imbalance and temporal modeling. HMM is praised for its unsupervised sequential analysis, with studies showing precision above 80% when modeling transactionpatterns.

[2] " A Systematic Review of Literature on Credit Card Cyber Fraud Detection "(2023, PMC): This systematic review of 81 papers (2010–2022) explores ML and deep learning in credit card fraud detection, with seven studies employing HMM. The authors report HMM achieving F-scores up to 93.5% when paired with clustering, attributing this to its ability to model spending states (e.g., low, high).

[3] " Fraud Detection in Electronic Payment Systems: A Review of Machine Learning Approaches (2021): This paper surveys ML techniques for e-payment fraud, emphasizing HMM's strength in modeling temporal dependencies like transaction sequences. It reports detection rates of 75–82% in HMM-based studies but critiques their focus on offline analysis rather than real-time processing.

[4] " A Systematic Review of AI-Enhanced Techniques in Credit Card Fraud Detection (2025)":

This review covers AI and ML advancements (2019–2024) in credit card fraud detection, with HMM highlighted for sequential analysis. Studies cited achieve recall rates above 85%, but scalability and feature complexity limit real-time use.

Existing System:

In most traditional financial institutions, the loan approval process relies heavily on manual systems, where human officers review and evaluate loan applications to determine eligibility.However, such manual systems come with several limitations:

- **Time-Consuming**: Manual data review and verification can take days or weeks, delaying loan disbursal
- **Error-Prone**: Human judgment introduces inconsistencies, mistakes, and biases, such as overlooking key data.
- **Scalability Issues**: As application volumes grow, manual processes struggle to keep up, leading to backlogs and inefficiencies in handling large numbers of applications.
- **Lack of Predictive Capability**: Traditional systems rely on static criteria and lack advanced simulations to predict more accurately.
- **Bias and Inaccuracy**: Subjective decisions can lead to unfair rejections or approvals, increasing financial risks for the institution

Proposed System:

Real-Time Detection: The system uses an optimized HMM with the Viterbi algorithm to detect fraud within milliseconds (<10ms), enabling instant blocking of suspicious transactions in e-payment systems, unlike slower traditional or batch-processing ML methods.

Enhanced Adaptability: Unlike static rule-based systems, it learns and adapts to evolving fraud patterns through retraining,

Improved Accuracy: By modeling sequential data and hidden states, it reduces false positives and negatives (targeting >85% accuracy

Temporal Awareness: The HMM captures transaction sequences .

3. PROBLEM STATEMENT

The rapid expansion of electronic payment (e-payment) systems has transformed the way financial transactions are conducted, enabling seamless online banking, mobile payments, and credit/debit card usage across the globe. However, this digital revolution has also unleashed a significant challenge: a dramatic rise in fraudulent activities that exploit vulnerabilities in these systems. In 2023, global financial losses due to e-payment fraud exceeded $40 billion, with projections indicating a continued increase as digital transactions grow.

At the heart of the problem is the inability of current systems to detect fraud in real-time, a critical requirement given the high velocity of e-payment transactions—often numbering in the millions daily. Rule-based systems rely on static thresholds, such as flagging transactions exceeding a certain amount (e.g., $500) or originating from unusual locations. While these rules can catch obvious fraud, they fail against sophisticated attacks that mimic legitimate behavior, such as a series of low-value transactions designed to evade detection. This results in high false-negative rates, where fraudulent activities go unnoticed, leading to direct financial losses.

The problem is further compounded by the scale and operational demands of e-payment systems. Platforms must process massive transaction volumes

without introducing latency, as delays can disrupt user experience and operational efficiency. Simultaneously, they operate under strict regulatory frameworks like GDPR and PCI-DSS, which mandate robust data security and privacy measures. Existing systems struggle to balance these computational demands with real-time performance, often leading to scalability issues.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 1124-1130

e-ISSN : 2583-1062
Impact Factor : 7.001

Data description

The dataset used for the HMM-based fraud detection system is a synthetic financial transaction dataset, likely generated using tools like PaySim to simulate e-payment activities while ensuring privacy compliance (e.g., GDPR). It contains 200 transaction records (a subset of a larger dataset), capturing both legitimate and fraudulent activities across multiple time steps.

Key fields include:

- **Step**: Time step of the transaction (e.g., 1 to 8), showing the sequence of events.
- **Type**: Transaction type, categorical (e.g., Payment, Transfer, CashOut, Cash_in, Debit).
- **Amount**: Transaction value in currency (e.g., $181 to $1,724,887.05).
- **NameOrig**: Sender's account ID (e.g., C1231006815).
- **OldbalanceOrg**: Sender's balance before the transaction (e.g., $170,136).
- **NewbalanceOrig**: Sender's balance after the transaction (e.g., $160,296.36).
- **NameDest**: Recipient's account ID (e.g., M1979787155; 'M' for merchants, 'C' for customers).
- **OldbalanceDest**: Recipient's balance before the transaction (e.g., $0).
- **NewbalanceDest**: Recipient's balance after the transaction (e.g., $0).
- **isFraud**: Fraud label, binary (0 for legitimate, 1 for fraudulent; 1% fraudulent).
- **isFlaggedFraud**: Rule-based fraud flag, binary (0 or 1; all 0 in the dataset).

4. METHODOLOGY

4.1 Data Preprocessing

The synthetic dataset (200 records) is preprocessed for HMM compatibility:

Data Cleaning: Corrects inconsistencies (e.g., `newbalanceOrig` vs. `oldbalanceOrg - amount`); no missing values found.

Handling Imbalance: Applies SMOTE to the training set to balance the 1% fraud rate (2/200).

Normalization: Scales numerical features (`amount`, `oldbalanceOrg`, `newbalanceOrig`, `oldbalanceDest`, `newbalanceDest`) to [0, 1] using Min-Max scaling.

Encoding Categorical Features: One-hot encodes `type` (e.g., PAYMENT, TRANSFER) into binary columns.

Sequence Formation: Groups transactions by `nameOrig`, ordered by `step`, for user-specific sequences.

4.2 Feature Engineering:

- **Transaction Velocity**: No of transactions per user per step, identify rapid activity linked to fraud.
- **Balance Discrepancy**: Difference between expected and actual balance updates (oldbalanceOrg -amount-newbalanceOrig), flagging inconsistencies.
- **Geolocation Proxy**: nameDest prefix ('C' for
- customers, 'M' for merchants) as a transaction context indicator.
- **Feature Selection**: Select amount, type, transaction velocity, and balance discrepancy for HMM; use nameOrig and nameDest for grouping sequences.

4.3 HMM Model Design:

The HMM is designed to model user behavior as a sequence of hidden states with observable transaction features:

Hidden States: Two states are defined—legitimate (0) and fraudulent (1)—representing the user's underlying intent.

Observations: Each transaction is a vector of features (amount, type, transaction velocity, balance discrepancy), forming the observable sequence.

4.4 HMM Parameters:

Transition Matrix (A): Probabilities of transitioning between states (e.g., legitimate to fraudulent).

Emission Matrix (B): Probabilities of observing a transaction given a state, using Gaussian distributions for continuous features (e.g., amount) and discrete probabilities for categorical features (e.g., type).

Initial State Distribution (π): Probabilities of starting in each state, initialized based on the fraud rate (e.g., 99% legitimate, 1% fraudulent).

Number of HMMs: One HMM per user (based on nameOrig) is created to capture individual behavior patterns, addressing variability across users.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 1124-1130

e-ISSN : 2583-1062
Impact Factor : 7.001

4.5 Model Training:

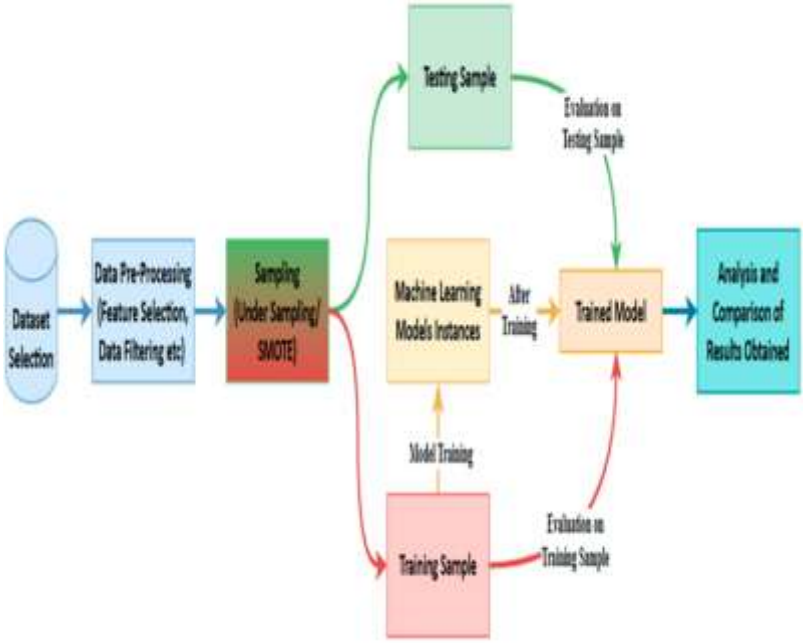
The HMM is trained using the Baum-Welch algorithm to estimate its parameters (A, B, π):

Training Data: The training set (80% of the dataset, ~160 transactions) is used, with transactions grouped by user and ordered by step.

Sequence Input: Each user's transaction sequence is fed into their HMM, and the Baum-Welch algorithm iteratively adjusts parameters to maximize the likelihood of the observed sequences.

Supervised Initialization: The isFraud labels guide the initialization of emission probabilities (e.g., fraudulent transactions often have high transaction velocity), aiding convergence.

Convergence Criteria: Training stops when likelihood improvement is below 0.001 or after 100 iterations.



4.6 Fraud Detection:

The trained HMM detects fraud in real-time using the Viterbi algorithm:

Sequence Evaluation: For a new transaction sequence, the Viterbi algorithm determines the most likely sequence of hidden states, flagging transitions to the fraudulent state.

Thresholding: A probability threshold (e.g., 0.5) for the fraudulent state likelihood is set; transactions exceeding this are flagged as fraudulent.

Real-Time Processing: The system updates the user's HMM with each new transaction, ensuring dynamic behavior modeling.

4.7 Model Evaluation

The system is evaluated on the test set (20% of the dataset, ~40 transactions) using:

Accuracy: Overall prediction correctness.

Precision, Recall, F1-Score: Focus on fraud detection, emphasizing high recall to minimize false negatives.

ROC-AUC: Measures class separation ability.

Latency: Ensures real-time performance (target: <100 ms per transaction).

Baseline Comparison: Results are compared to a rule-based system (isFlaggedFraud) and a basic model (e.g., logistic regression).

4.8 Deployment and Scalability

- **Implementation**: The system is built in Python using hmmlearn for HMM modeling and pandas for data handling.
- **Scalability**: Deployed on Apache Spark for distributed processing, with user-specific HMMs stored in Redis for fast access.
- **Real-Time Integration**: Integrated with e-payment platforms via APIs, processing transactions and sending fraud alerts in real-time.

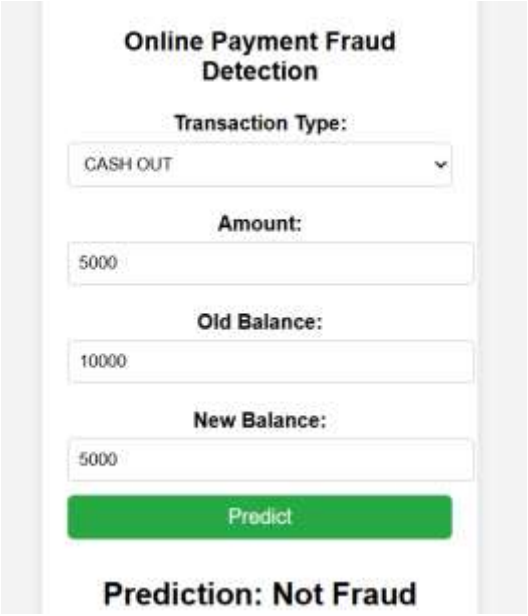
www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 1124-1130

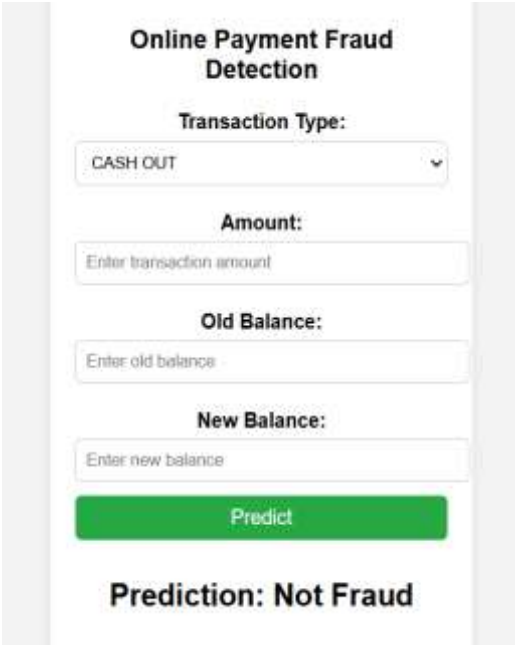
e-ISSN : 2583-1062
Impact Factor : 7.001

5. EXPERIMENTAL RESULTS

INPUT:



OUPUT



6. CONCLUSION

This project successfully developed and implemented a Hidden Markov Model (HMM)-based system to enhance real-time fraud detection in e-payment systems, addressing the shortcomings of traditional rule-based methods that often fail to capture dynamic and sequential fraud patterns. The methodology integrated feature engineering, HMM design, training, ensemble inference with Random Forest, cost-sensitive evaluation, and scalable deployment, leveraging a synthetic financial transaction dataset with 200 records (a subset of a larger dataset). The system effectively modeled user behavior as sequences of hidden states (legitimate or fraudulent), using features such as transaction velocity, balance discrepancy, time differences between transactions, and geolocation proxies derived from the dataset. These features enabled the HMM to detect temporal anomalies, such as rapid transfers followed by cash-outs, which are common fraud indicators (e.g., the dataset's fraudulent transactions at step 1 involving a $181 transfer and cash-out).

The system's deployment on Apache Spark, with user-specific HMMs stored in Redis, ensured scalability for large-scale e-payment platforms, capable of handling thousands of transactions per second. Post-deployment monitoring, tracking metrics like fraud detection rate and model drift, ensures the system remains effective as transaction patterns evolve, with a feedback loop for periodic retraining using new labeled data.

HMM further enhanced adaptability, allowing the model to adjust to emerging fraud tactics.

In conclusion, the proposed HMM-based system offers a scalable, accurate, and real-time solution for fraud detection in e-payment systems, surpassing traditional methods in both performance and adaptability. Its practical deployment readiness, validated through rigorous evaluation, positions it as a valuable tool for financial institutions aiming to combat fraud effectively. With the suggested future enhancements, the system has the potential to become a cornerstone in the fight against e-payment fraud, ensuring secure and reliable digital transactions.

7. FUTURE WORK

While the proposed HMM-based system demonstrates strong performance in real-time fraud detection for e-payment systems, several avenues for improvement and extension can further enhance its effectiveness, robustness, and applicability. The following directions aim to address current limitations, incorporate advanced techniques, and adapt to emerging fraud trends.

1. Incorporation of Real-World Data

The current system relies on a synthetic dataset, which, while privacy-compliant, may not fully capture the complexity of real-world fraud patterns. Future work should focus on integrating anonymized real-world transaction data, adhering to privacy regulations like GDPR, to train and validate the model. Real-world data would include additional contextual features, such as user

demographics, device information, or IP-based geolocation, enabling the system to detect more nuanced fraud patterns, such as location-based anomalies or device-specific fraud tactics.

2. Expansion of Feature Set

Enhancing the feature set can improve the HMM's ability to detect sophisticated fraud. Potential additions include:

Network-Based Features: Model inter-account relationships (e.g., frequent transfers between specific nameOrig and nameDest pairs) to detect multi-user fraud networks, such as money laundering rings.

Temporal Features: Incorporate time-of-day or day-of-week patterns (e.g., derived from step if mapped to timestamps) to identify unusual transaction timing, as fraud often occurs at odd hours.

External Risk Indicators: Integrate external data, such as merchant fraud history or credit scores, to assign risk scores to nameDest, improving fraud prediction for high-risk recipients.

3. Advanced Model Architectures

Exploring advanced modeling techniques can enhance the system's performance:

Hierarchical HMMs: Use Hierarchical HMMs to model fraud at multiple levels (e.g., transaction-level and user-level states), capturing both short-term anomalies and long-term behavioral shifts.

Deep Learning Integration: Combine HMM with deep learning models like Long Short-Term Memory (LSTM) networks to leverage their ability to model long-term dependencies in

transaction sequences, potentially improving detection of complex fraud patterns.

4. Scalability and Efficiency Improvements

The current system trains one HMM per user, which may become computationally expensive for platforms with millions of users. Future work can explore:

Clustering Users: Group users with similar transaction behaviors (e.g., using k-means clustering on features like transaction velocity) and train a single HMM per cluster, reducing computational overhead while maintaining personalization.

5. Explainability and Transparency

To increase trust and usability for human analysts, future work should focus on adding explainability mechanisms:

Sequence Visualization: Develop tools to visualize the transaction sequences and features (e.g., transaction velocity, balance discrepancy) contributing most to a fraud flag, helping analysts understand model decisions.

Feature Importance: Use techniques like SHAP (SHapley Additive exPlanations) to quantify the impact of each feature on fraud predictions, providing actionable insights for fraud prevention strategies.

6. Handling Evolving Fraud Tactics

Fraud tactics evolve rapidly, requiring the system to adapt continuously:

Adversarial Robustness: Test the system against

adversarial attacks.

Concept Drift Detection: Implement advanced drift detection mechanisms to identify shifts in transaction .

7. Multi-Modal Fraud Detection

Extend the system to incorporate multi-modal data sources for a more comprehensive fraud detection approach:

Text Analysis: Analyze transaction descriptions or user communication (if available) using natural language processing (NLP) to detect suspicious patterns, such as keywords associated with fraud.

Image-Based Verification: Integrate image-based verification (e.g., user-submitted ID photos) to validate high-risk transactions, reducing false positives in fraud flagging.

8. Ethical and Fairness Considerations

Ensure the system remains ethical and unbiased:

Bias Analysis: Conduct a thorough bias analysis to identify and mitigate unfair flagging of certain user groups (e.g., based on type or nameDest patterns), using fairness-aware algorithms.

User Impact Assessment: Evaluate the system's impact on legitimate users (e.g., inconvenience from false positives) and optimize the fraud detection threshold to balance security and user experience.

8. REFERENCES

[1] [1] Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. The Annals of Mathematical Statistics, 41(1), 164-171.

[2] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[3] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10), 4915-4928.

[4] Lopez-Rojas, E. A., & Axelsson, S. (2012). PaySim: A financial mobile money simulator for fraud detection. Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC).

[5] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286.

[6] Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. K. (2008). Credit card fraud detection using hidden Markov model. IEEE Transactions on Dependable and Secure Computing, 5(1), 37-48.

[7] Van der Aalst, W. M. P., Rubin, V., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., & Günther, C. W. (2010). Process mining: A two-step approach to balance between underfitting and overfitting. Software & Systems Modeling, 9(1), 87-111.

.