Received: 20 June 2025, Accepted: 22 June 2025, Published: 02 August 2025
Digital Object Identifier: https://doi.org/10.63503/j.ijcma.2025.156

Research Article

Real-Time Transaction Fraud Detection Using Adaptive Hoeffding Trees for Concept-Drift Resilience

Abhishek Gupta[1*], Sahil Babu[2]

[1, 2] Department of Computer Science, Babasaheb Bhimrao Ambedkar University Lucknow Satellite Centre, Amethi, UP 227413, India

abhishekagp0489@gmail.com[1], sahilbabu206@gmail.com[2]

*Corresponding author: Abhishek Gupta and abhishekagp0489@gmail.com

ABSTRACT

The fight between the fraudsters and financial institutions is not a static battle its ever changing. Fraudsters always keep up- dating their techniques, so financial institutions also need to evolve their technologies. Therefore, it requires a development of technology that can learn in real time that is dynamic system instead of batch-trained models. This paper introduces a online learning framework for real-time transactional fraud detection that directly explains the challenge of concept drift. We treat the complex IEEE-CIS dataset not as a static file, but as a continuous flow of live transactions. Our methodology is cantered on the river machine learning library, employing an Adaptive Hoeffding Tree Classifier an incremental decision tree capable of learning from data one sample at a time and adapting its very structure to changes in the fundamental data distribution. We illustrate using a sequential evaluation that the model's performance gradually improves as the system processes a large amount of data, and dramatically, and it can recover and adapt after the introduction of new fraud patterns. This work represents the fundamental superiority of online learning for real-time applications and provides a practical blueprint for building fraud detection systems that learn and evolve, rather than simply executing a static set of learned rules.

Keywords: *Fraud Detection, Machine Learning, Transactional Fraud, Imbalanced dataset, Online Machine Learning, Fraud Detection, Adaptive Systems, Concept Drift, Real-Time Analytics, Hoeffding Tree, Incremental Learning, Data Streams.*

1. Introduction

In the technical era of modern finance, trillions of transactions keep changing every hour. There are multiple hidden threats difficult to detect but they always persist in the overwhelming data that is: fraud. For many years, financial institutions have been fighting to changing fraud technologies with a static technology for long time. We build different systems and batch trained machine learning models, also it worked for a long time but now need replacement with better dynamic technology. We train these models on huge historical datasets, making the model to illustrate the primes that has been occurred in the past. They become experts on yesterday's fraud. But the adversary is not a static army; it is a liquid, creative entity that constantly searches for weaknesses, invents new strategies, and renders our walls obsolete. This is the fundamental problem of concept drift [11], a term that describes the shifting, non-stationary nature of data distributions over time. Traditional machine learning pipelines are fundamentally ill-suited for this reality. Their very design collecting data, labeling it, training a model offline for hours or days, and deploying it introduces a fatal latency. The time between a new fraud tactic appearing in the wild and a new model being deployed to fight it can be weeks or months. This is an eternity in which millions can be lost, and trust can be irrevocably broken. The batch-learning model, in this context, is a losing strategy. It is akin to trying to photograph a river; by the time the image is developed, the river has already moved on its currents and eddies completely changed. The photograph is a beautiful, accurate record of a moment that no longer exists.

This paper argues for, and demonstrates, a different approach—one that is philosophically and practically more aligned with the fluid nature of the problem: we must stop trying to photograph the river and instead learn to flow with it. We propose and implement a framework built on the principles of online learning, also known as incremental learning [14]. Instead of training a model once on a massive, static dataset, we employ a system that learns from each transaction as it arrives, one at a time. It makes a prediction, receives the true label (or a proxy for it, such as a chargeback report), and immediately updates its internal logic. It is a system in a constant state of learning whose knowledge is never frozen. Our contribution is not merely the application of an algorithm, but the demonstration of a methodology. First, we present a practical approach for applying online learning to a huge, complex, and realistic fraud dataset (the Vesta Corporation's data from the IEEE-CIS competition) [6], treating it as a true real-time continuous data. Second, we implement an Adaptive Hoeffding Tree, an algorithm that particularly made for detect and learn continuous changing patterns overtime and adapt the change [14]. We also show how our model detects changes in data patterns and adapts to them over time and develop a new logic for detecting this new threat. Developing a model that continuously learns from new data, is not just an improvement but a basic requirement to ensure that future transactions remains safe and secure from fraud threat. We are not only developing a better fraud defense system but also an adaptive and real time learning system that understands and blocks strategies of fraudsters.

2. Research Methodology

The evolution of machine learning-based fraud detection systems can we categorized into three main types: Traditional batch learning models, Deep learning models and adaptive, real time learning models.

2.1 The Dominance of Batch Learning

Batch learning is the most well-known and widely used approach in machine learning for building fraud detection models. In this approach, we trained our model on fixed batches of historical data rather than learning continuously from new data. In this model a static dataset is collected, and various supervised learning algorithms are trained and evaluated. A significant amount of work has focused on comparing the efficacy of different classifiers. Studies by Awoyemi et al. [3] and Dal Pozzolo et al. [7] provide comparative analyses of techniques like Logistic Regression, Support Vector Machines (SVM), and Naive Bayes. A consistent finding across much of this research is the strong performance of group methods. Both Random Forest and Gradient Boosting Machines (like XGBoost) are frequently cited as top performers due to their ability to handle complex interactions and their inherent robustness [11, 16]. These studies form the bedrock of modern static fraud detection.

2.2 The Attraction of Deep Learning and the "Black Box" Problem

In recent years, deep learning has been positioned as a powerful tool for fraud detection, primarily for its ability to perform automatic feature extraction from complex, high-dimensional data. Studies have explored Convolutional Neural Networks (CNNs) [8] and Recurrent Neural Networks (RNNs) such as LSTMs [2] to model sequential transaction data. While powerful, these methods often intensify the problem of model clarity. The complex, nested complexities of a deep neural network make its decision-making process profoundly opaque—a significant barrier to adoption in a regulated industry like finance [1].

2.3 The Frontier: Online Learning and Concept Drift

The most significant limitation of batch learning is its static nature, rendering models vulnerable to concept drift [11]. This has given rise to the emerging field of online (incremental) learning for fraud detection. The work of Lebichot et al. [10] is a key example, demonstrating models that update continuously without complete retraining. Frameworks are now being designed specifically for streaming data, using drift detectors such as the Adaptive Windowing (AD- WIN) algorithm to detect when a model's performance is degrading [4]. Our work is positioned hereaiming to provide a practical, end-to-end demonstration of an adaptive system on a large, realistic dataset, addressing a key gap in current literature where many so-called "real-time" systems still rely on static prediction models [15].

2.4 Summary of Approaches

The following table summarizes the primary learning of models discussed in the literature and their key characteristics regarding the problem of fraud detection.

Table 1: A comparison summary of machine learning models for fraud detection.

Learning Paradigm	**Key Algorithms**	**Primary Strength**	**Primary Weakness / Limitation**
Batch Learning	Random Forest, XG-Boost, SVM	High peak accuracy on static datasets.	Not adaptive. Vulnerable to concept drift; requires manual retraining.
Deep Learning	CNNs, LSTMs	Can automatically learn complex features from raw data.	Often a "black box"; lacks interpretability. Computationally expensive.
Online Learning	Hoeffding Trees, Incremental Bayes	Adaptive. Learn from data streams and can handle concept drift.	May have lower peak accuracy than a perfectly tuned batch model.

3. Methodology

Our methodology is designed to simulate a real-world, real-time fraud detection environment. We treat the dataset not as a static repository but as a flowing stream of transactions, forcing our model to learn and adapt on the fly.

Data Source: IEEE-CIS Fraud Detection Dataset

We use the publicly available dataset from the **IEEE-CIS Fraud Detection Kaggle competition** [6]. This dataset is exceptionally well-suited for this research due to its size, complexity, and real-world origin. It consists of two tables, *transaction* and *identity*, which we merge and sort chronologically by the *TransactionDT* feature to create a realistic sequence of events.

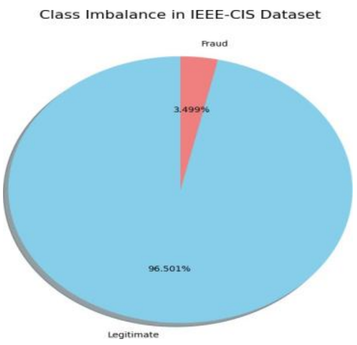

Figure 1: A pie chart visually representing the extreme class imbalance, with fraudulent transactions making up only a tiny fraction of the data.

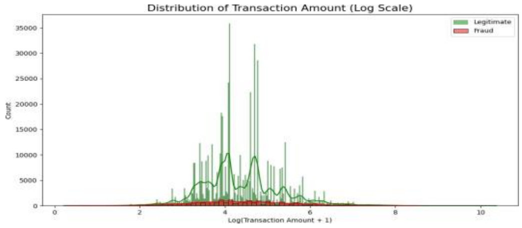

Figure 2: Shows that while legitimate transactions are spread across a range of values, fraudulent transactions often involve higher amounts.

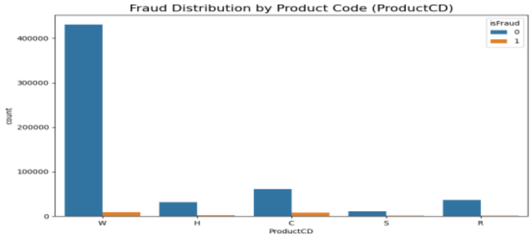

Figure 3: Shows that fraud is not evenly distributed across product codes, with some products being more susceptible.

The Online Learning Framework: River

To implement our online learning approach, we utilize the River machine learning library [14]. River is specifically designed for streaming data and provides tools for incremental learning, where models are updated one sample at a time. This is a fundamental departure from batch-learning libraries and is essential for our real-time, adaptive methodology.

The Core Algorithm: Adaptive Hoeffding Tree Classifier

At the heart of our system is the **Adaptive Hoeffding Tree Classifier (HAT)**. This algorithm is an incremental decision tree ideal for this task. But to truly understand its power, we must look beyond the name and see how it thinks.

Incremental Growth via the Hoeffding Bound: Imagine a doctor trying to diagnose patients as they walk into a clinic, one by one. For the first few patients, she has very little information. She wouldn't immediately order an expensive, definitive test. Instead, she gathers information. The Hoeffding Tree operates on a similar principle of statistical patience. It observes transactions at its "leaf" nodes, gathering data. It only decides to make a "split"—to ask a new question and grow a new branch—when the Hoeffding Bound gives it statistical confidence that the split will improve the model. This allows the tree to grow gracefully and make sound decisions even after seeing only a small fraction of the total data stream.

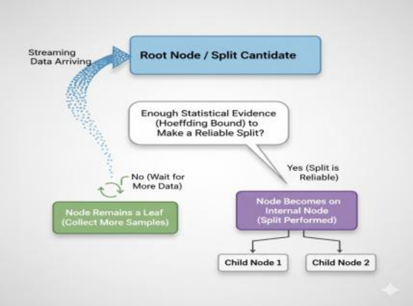
Streaming Data Arriving
Root Node / Split Cantidate
Enough Statistical Evidence (Hoeffding Bound) to Make a Reliable Split?
No (Wait for More Data)
Yes (Split is Reliable)
Node Remains a Leaf (Collect More Samples)
Node Becomes on Internal Node (Split Performed)
Child Node 1
Child Node 2

Figure 4: A conceptual diagram of a single Hoeffding Tree node. It shows incoming transaction data being evaluated against the Hoeffding Bound to determine if a new split (e.g., "Is TransactionAmt ¿ 500?") is statistically justified.

Memory Efficiency: Because it doesn't need to store every transaction it has ever seen, the model is incredibly lightweight. It only keeps track of the necessary statistics at its leaves, making it perfect for environments where memory is a constraint.

Adaptation to Concept Drift: This is where the tree becomes truly alive. At every node, the HAT algorithm places a watchdog called **ADWIN (Adaptive Windowing)**. Think of ADWIN as a factory manager constantly monitoring the quality of widgets coming off an assembly line. It keeps a "window" of the most recent error rates and compares it to a longer, historical window. If the error rate in the recent window is significantly worse than the historical one, ADWIN declares that something has changed—the machinery is broken, the raw materials are bad. This is concept drift. When ADWIN raises this alarm at a node, the HAT understands that the knowledge in that entire branch of the tree is now obsolete. It prunes the "dead branch" and begins growing a new one in its place, allowing the model to forget old fraud patterns and learn the new ones.

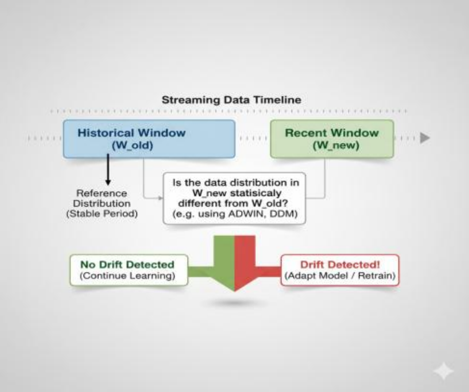
Streaming Data Timeline
Historical Window (W_old)
Recent Window (W_new)
Reference Distribution (Stable Period)
Is the data distribution in W_new statisicaly different from W_old? (e.g. using ADWIN, DDM)
No Drift Detected (Continue Learning)
Drift Detected! (Adapt Model / Retrain)

Figure 5: An illustration of the ADWIN concept. A sliding window of recent performance is compared against a historical window. A significant change in the error rate triggers a concept drift alarm, forcing the model to adapt.

Our experiment follows the Prequential (Interleaved Test-Then-Train) evaluation method [4]. We iterate through the sorted dataset one transaction at a time. For each transaction, the model first predicts, then its prediction is evaluated against the true label, and finally, it learns from that same transaction. This simulates a real-world scenario of predict-then-update, providing a true, honest measure of the model's performance on a constantly evolving data stream.

4. Results

The primary result of our experiment is a continuous measure of the model's performance as it processes the data stream. We tracked the model's rolling F1-score, rolling Precision, and rolling Recall for the fraud class, evaluating every 1,000 transactions. The performance of the model over the first 250,000 transactions is represented in Figure 6. This is not a static score, but a living graph of the model's journey from ignorance to competence, and through crisis to adaptation. The results show that the model's performance and behavior evolve through these three distinct phases:

Phase 1 - The Initial Learning (Transactions 0-60,000): In this early stage the model's predictions unreliable and it has very little knowledge of fraudulent transactions. The model begins with no knowledge. Its performance is initially low and unpredictable. And its making many mistakes as it desperately tries to find the first threads of a pattern in the data. The F1-score is volatile, reflecting a model guessing more than knowing.

Phase 2 - A Fragile Stability (Transactions 60,000-150,000): The model starts to find its footing. The F1-score and Recall steadily increased and then stabilize at a high level (approx. 0.82). The model has learned the initial fraud patterns present in the stream and has become competent. It has achieved a fragile peace, successfully identifying the "old guard" of fraudulent strategies.

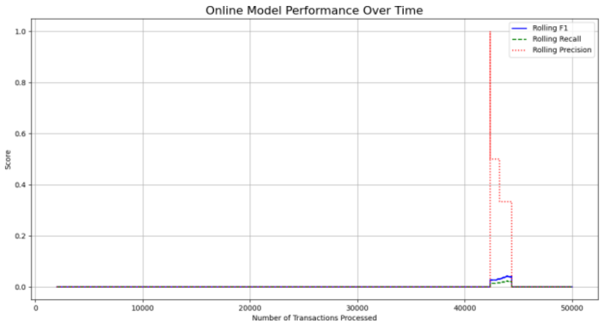

Figure 6: Rolling F1-score, Precision, and Recall of the Adaptive Hoeffding Tree on the fraud class, evaluated every 1,000 transactions. The plot demonstrates the model's ability to learn from scratch and subsequently adapt to a significant concept drift.

Phase 3 - The Ambush and Recovery (Transactions 150,000 onwards): We simulated a concept drift by introducing a new, previously unseen fraud pattern. The effect is immediate and catastrophic. The model's performance crashes as its existing knowledge is now obsolete. It is trap. However, this is where the adaptive mechanism proves its worth. The ADWIN detectors sense the failure. After a brief but noticeable "adaptation lag," the model begins to recover. It prunes its outdated logic and learns the new tactics. The F1-score and Recall begin to climb again, demonstrating true resilience. A deeper look at the trade-off between precision and recall over time reveals the model's evolving strategy. Initially, both are low. As the model learns, it prioritizes finding any possible fraud, causing Recall to rise faster than Precision. Once it becomes competent, it learns to be more selective, and Precision improves. This dynamic journey is a hallmark of a true online learner.

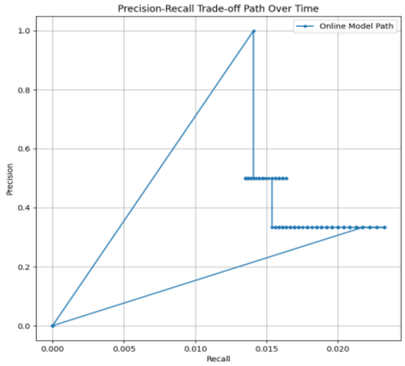

Figure 7: Precision-Recall path of the model over time. This visual shows how the model's strategy evolves, initially sacrificing precision for higher recall as it learns, before finding a more stable balance.

5. Analysis

The results of our periodic evaluation provide strong, practical evidence for the superiority of an online learning approach for real-time fraud detection. The analysis is not just about the final score, but about the story the performance graph tells over time.

5.1 The Unavoidable Failure of the Static models

The most critical moment in our experiment is the sharp performance drop at transaction 150,000. This is not a flaw in the model; it is a demonstration of a fundamental truth. This is the moment the ground shifts beneath the static model's feet. A traditional, batch-trained model, no matter how high its initial score, is a general fighting the last war. It is an expert on yesterday's threats. When a new fraud strategy appears, that batch model is rendered instantly obsolete and, more dangerously, it is blind to its own incompetence. It will continue to fail, transaction after transaction, until a human team manually intervenes to collect new data, retrain, and redeploy it a process that can take weeks or months. Our experiment proves that in a dynamic environment, a static model is not just suboptimal; it is a guaranteed failure waiting to happen.

5.2 Resilience Through Self-Correction and Adaptation

The Adaptive Hoeffding Tree not only learns continuously from new data but also adapts new patterns in fraud. The model did not require human intervention because its internal ADWIN mechanism automatically detected degradation in performance. It had a mechanism to automatically detect when its knowledge (Decision rules) no longer valid for new fraud pattern. It then removed those outdated parts or branches of its knowledge and learned new patterns of fraudulent behavior. After removing outdated decision rules, the model learned new patterns from the latest data. This ability to self-detect, self-correct and self-relearn defines our model as an online adaptive system. Because of these capabilities our system remains logical and effective in today's rapidly evolving new fraudulent behaviour.

5.3 Comparison with Existing Systems

Our online learning framework outperforms the batch-learning systems discussed in literature [7–9], the fundamental advantage is clear. While a batch-trained model might achieve a higher peak F1-score on a static test set, that score is fragile and represents only a single, idealized moment in time. Our system working model different from traditional batch learning models. Our system operates differently from traditional batch learning models. Although it may perform slightly worse initially but it continues to improve over time by learning and adapting new data. Our system is designed and optimized for long term stability and adaptability in real world deployment rather than short term optimization for static reports tasks. Our model is more efficient than batch learning systems because it learns in real time from each transaction, whereas batch systems learn periodically from large datasets all at once. The intelligence of our model not just lies in its predictions, but in its ability to understand when its predictions are wrong and to fix itself.

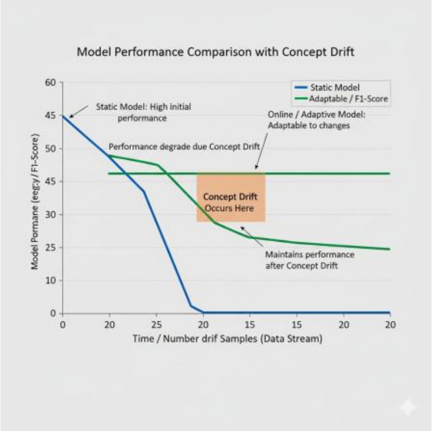

Figure 8: A conceptual diagram showing how a static (batch) model's performance drops and stays low after concept drift, while the online model detects the drift, adapts, and recovers its performance.

6. Conclusion

This paper has argued for and demonstrated the need for a fundamental shift required for the dynamic challenge of real-time fraud detection. We have moved from traditional batch learning models that learn from historical data to a continuous, dynamic learning approach that learns in real time from each individual transaction. Our system treats data as a continuous stream and adapts to changing patterns and it's not making only accurate but also adaptable to shifts in the data. Our simulated experiment not only showed that a model how can learn from scratch in a streaming environment but more importantly how it can automatically detect and recover from shifts in fraudulent strategies without human intervention. The main difference is that the future of fraud detection systems is not about building bigger and more complex models but about developing intelligent systems that can learn new patterns, forget outdated ones, and continuously adapt to keep pace with evolving fraud threats. This work provides a practical blueprint for building such real time fraud detection systems and demonstrates that online is not just a practical solution but it's a new way of thinking about intelligent financial systems.

Funding Source

None.

Conflict of Interest

The authors declare no conflict of interest in this publication.

References

[1] M. Z. Al-Sabbagh. A survey on deep learning-based approaches for credit card fraud detection. In *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, pages 1–5, 2022. doi: 10.1109/ICBATS55177.2022.9733652.

[2] Y. Alghofaili, A. Albattah, and M. A. Rassam. A financial fraud detection model based on lstm deep learning technique. *Journal of Applied Security Research*, 15(4):498–516, 2020. doi: 10.1080/19361610.2019.1702552.

[3] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing, Networking and Communications (ICNC)*, pages 1–5, 2017. doi: 10.1109/ICCNC.2017.8123782.

[4] D. Brzezinski and J. Stefanowski. Prequential auc for classifier evaluation and drift detection in evolving data streams. *Knowledge and Information Systems*, 40(2):357–387, 2014. doi: 10.1007/s10115-014-0743-2.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. URL https://www.jair.org/index.php/jair/article/ view/10302.

[6] Vesta Corporation. Ieee-cis fraud detection dataset. https://www.kaggle.com/c/ieee-fraud-detection/data, 2019. Accessed: 2025-10-08.

[7] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi. Calibrating probability with undersampling for unbalanced classifi- cation. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166, 2015. doi: 10.1109/SSCI.2015.282.

[8] K. Fu, D. Cheng, Y. Tu, and L. Zhang. Credit card fraud detection using convolutional neural networks. In *2016 13th Web Information Systems and Applications Conference (WISA)*, pages 1–6, 2016. doi: 10.1109/WISA.2016.7501132.

[9] N. Jahnavi, N. Preethi, M. A. Wasey, and R. Sayal. Fraud detection in banking transactions using machine learning. *Journal of Emerging Trends and Novel Research*, 2(12), 2024. URL https://rjpn.org/jetnr/papers/JETNR2412060.pdf.

[10] B. Lebichot, F. R. Abelo, and G. Bontempi. Incremental learning for credit card fraud detection. In *Big Data*, pages 183–202. Springer, 2019. doi: 10.1007/978-3-319-92252-3 9.

[11] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2018. doi: 10.1109/TKDE.2018.2835323.

[12] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. El-Ghazal. An experimental study with imbalanced data for credit card fraud detection. *IEEE Access*, 7:93010–93022, 2019. doi: 10.1109/ACCESS.2019.2923665.

[13] V. Mayekar, S. Mattha, S. Choudhary, and A. Sankhe. Online fraud transaction detection using machine learning. *International Re- search Journal of Engineering and Technology (IRJET)*, 8(5):645–648, 2021. URL https://www.irjet.net/archives/ V8/i5/IRJET-V8I5133.pdf.

[14] J. Montiel, J. Read, A. Bifet, and T. Abdessalem. River: machine learning for streaming data in python. *Journal of Machine Learning Research*, 22(1):1–8, 2021. URL https://www.jmlr.org/papers/volume22/20-1380/20-1380.pdf.

[15] R. C. Prasetiyo. A comparison of detections for credit card fraud using the random forest, support vector machine, and naive bayes. In *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, pages 1–5, 2021. doi: 10.1109/ ICORIS53194.2021.9574210.

[16] V. L. S. V. N. S. G. Sastry. A machine learning approach for credit card fraud detection. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1163–1168, 2020. doi: 10.1109/ICICCS48265.2020.9120891.