Credit card fraud classification using applied machine learning – a comparative study of 24 machine learning algorithms

Kelechi K. Amamba[1,*], *Olufemi S. Oloniluyi*[1], *Olayinka H. Sikiru* [1]

[1]Department of Information Systems and Business Analytics, Kent State University, Kent, Ohio 44242, USA.

*Corresponding Author: Kelechi K. Amamba (Email: kamamba@kent.edu)
(Received: 19-August-2025; accepted: 12-November-2025; published: 31-December-2025)
http://dx.doi.org/10.55579/jaec.202594.515

Abstract. *This paper presents a comprehensive study on credit card fraud detection, addressing the escalating issue of fraudulent activities that significantly impact both financial institutions and consumers. We introduce a novel framework for evaluating the collective performance of diverse machine learning (ML) models—including Logistic Regression, Decision Trees, Random Forests, Support Vector Machines, and Neural Networks—using a synthetic dataset carefully constructed to mirror real-world transaction features and behavioral patterns. By applying various sampling strategies to this highly imbalanced dataset and leveraging domain knowledge for feature selection, this study aims to enhance both the accuracy and stability of fraud detection models, while identifying the minimum feature set required for optimal detection speed and efficiency. Our results reveal that algorithms such as Gaussian Naive Bayes, Kernel Naive Bayes, Cubic SVM, and Trilayered Neural Networks each provide strong, balanced performance. Building on these findings, we propose that ensembling these top-performing models could further improve detection rates and reliablity, harnessing their complementary strengths to achieve superior overall performance. This paper underscores the necessity of advanced and integrated ML techniques for robust, timely fraud detection, offering valuable insights for real-time implementation and presenting a comprehensive solution to a pressing financial security challenge.*

Keywords: *Applied machine learning; Credit card fraud detection; Ensemble learning in financial security; Feature selection for fraud detection; Machine learning fraud models.*

1. Introduction

Credit card fraud is a growing global threat to financial institutions, consumers, and the economy. Losses reached $35 billion in 2023, reflecting both rising cases and the increasing complexity of detection [1]. Beyond financial damage, millions of consumers experience significant distress each year. In the U.S., the FTC recorded about 440,000 fraud reports in 2023 [2], making it one of the nation's most common financial crimes [3].

As the leading form of identity theft, credit card fraud affects 65% of U.S. cardholders, with nearly half targeted multiple times [1]. For every dollar lost, financial institutions spend roughly three more on investigations, disruptions, and mitigation [1]. These costs often translate into higher fees and interest rates for consumers,

while the ensuing reputational damage undermines trust and competitiveness.

Looking ahead, U.S. fraud losses could exceed $165 billion within the next decade, emphasizing the urgent need for advanced, reliable detection systems [3]. Effective solutions must detect fraud in real time while minimizing false positives that erode consumer confidence.

Recent data highlights the scale of the problem: global fraud losses were projected to surpass $43 billion in 2024, with the U.S. accounting for nearly 46% [1]. The FTC continues to rank credit card fraud among the top forms of identity theft [2]. Densely populated states such as California and Florida report especially high incidences, driven by large transaction volumes, and require heightened vigilance and more sophisticated detection methods [1].

To address these challenges, this research provides practical insights into reducing fraud, improving detection speed, and strengthening the security of credit card transactions. By mitigating risks effectively, financial institutions can protect their assets while boosting customer confidence in an increasingly digital and vulnerable financial landscape. The study compares advanced machine learning techniques—Logistic Regression, Decision Trees, Random Forests, SVMs, and Neural Networks—using realistic, imbalanced transaction datasets. It also explores feature selection strategies and the role of domain knowledge in improving the accuracy, precision, and efficiency of fraud detection models.

The machine learning models in this study were trained and optimized in MATLAB using the Ohio Supercomputer Center's advanced computational capabilities. A key contribution is the systematic evaluation of multiple models to identify effective combinations for fast, reliable fraud detection. Preliminary results highlight ensemble approaches—combining Gaussian Naive Bayes, Kernel Naive Bayes, Cubic SVM, and Trilayered Neural Networks—as promising for improving detection accuracy, speed, and misclassification management.

The primary methodological novelty of this study is an Adaptive Data Handling Framework that integrates (1) stratified and reproducible sampling, (2) dual feature-selection experiments—A-Test (automated, statistics-driven) and B-Test (domain-refined)—and (3) model-aware preprocessing involving normalization and weighting tuned per algorithm. Unlike prior studies that primarily benchmark algorithms on the Kaggle dataset using standard scaling or oversampling, this framework explicitly merges domain knowledge with statistical feature selection, quantifies its impact through a controlled design-of-experiments approach (identical data splits across A/B tests), and evaluates models using cost-sensitive metrics. The study's originality therefore lies not in the dataset itself, but in the data-handling and evaluation protocol, which enhances robustness, interpretability, and deployment relevance for modern financial fraud detection systems.

2. Background

Credit card fraud detection is an increasingly critical issue confronting both financial institutions and consumers due to rapidly evolving fraudulent techniques that consistently outpace traditional detection methods. Fraudulent activities in credit card transactions have seen significant growth, driving the demand for innovative and proactive detection strategies [4]. Machine Learning (ML) has become a crucial component in addressing these challenges, proving effective in detecting anomalous patterns and significantly mitigating financial losses [4]. This literature review synthesizes current research and various machine learning techniques applied to credit card fraud detection, highlighting complexities and challenges associated with effective fraud mitigation.

2.1. Landscape of Credit Card Fraud:

Traditional fraud detection approaches, largely reliant on static rule-based systems, have struggled to keep pace with increasingly sophisticated and dynamic fraudulent strategies. These rule-based systems, while initially effec-

tive, have limitations in adapting to new types of fraudulent activities and can lead to significant financial losses if undetected frauds occur [5]. The shift towards more adaptive and dynamic machine learning methods addresses these challenges by continually updating detection models based on evolving transaction patterns [6]. The continuous adaptation to new fraud trends and behaviors underscores the significance of integrating machine learning techniques into fraud detection frameworks.

2.2. The Role of Feature Engineering:

Effective fraud detection systems rely heavily on comprehensive feature engineering to capture essential patterns and behaviors indicative of fraud. Features commonly utilized include transaction amounts, merchant categories, geographic locations, temporal aspects, and historical transaction data [7]. However, domain knowledge integration significantly enhances feature engineering by guiding the selection of features. This advanced preprocessing technique systematically reduces irrelevant or redundant features, further increasing model accuracy and efficiency [7] [8] [9].

2.3. Addressing Data Imbalances:

One persistent challenge in fraud detection is managing imbalanced datasets, as fraudulent cases represent a minor fraction compared to legitimate transactions. Techniques such as Synthetic Minority Over-Sampling Technique (SMOTE) and cost-sensitive learning methods are extensively used to handle class imbalance, thus enhancing model performance and reducing bias towards majority classes [10] [11].

2.4. Real-Time Processing and Interpretability:

The necessity for real-time fraud detection prompts integration of streaming data processing with machine learning models, emphasizing rapid response without compromising accuracy [4]. However, the complexity of advanced models like deep neural networks introduces challenges regarding model interpretability. Explainable Artificial Intelligence (XAI) has emerged to enhance model transparency and build trust in automated systems, aligning with regulatory compliance and ethical standards [12].

This comprehensive review underscores the essential role of machine learning and feature engineering strategies in advancing credit card fraud detection, highlighting the importance of continual adaptation and integration of new methodologies to address evolving fraud detection challenges.

3. Method

3.1. Dataset Preparation and Class Imbalance Handling Strategy:

The dataset used was obtained from Kaggle [13] and comprises anonymized credit-card transactions recorded between 1 January 2019 and 31 December 2020, totaling 555,719 transactions across 800 merchants, with both legitimate and fraudulent entries. Although personal attributes such as age, gender, and date of birth are included, all identifiers were anonymized in accordance with Kaggle's data-use policy. Because such variables may reflect sensitive demographic characteristics, analyses were conducted with careful attention to fairness and without any attempt at re-identification. The research avoided reinforcing stereotypes or introducing bias in model interpretation, thereby aligning with principles of privacy, transparency, and equitable treatment of individuals represented in the data.

Data preparation is a crucial phase that involves cleaning, transforming, and organizing raw data into a format suitable for analysis and model development. This process ensures the quality, consistency, and relevance of the data—factors essential for building reliable machine-learning models [14]. The present

study followed a reproducible, model-aware pipeline referred to as the Adaptive Data Handling Framework. This framework, implemented identically across both A-Test and B-Test, ensures that any observed performance differences derive solely from feature-selection strategies rather than random variation in sampling or preprocessing.

Prior to modeling, comprehensive integrity checks were performed. Variable types were verified to ensure that each attribute was correctly classified as numerical, categorical, or date-time, following best practices outlined by Wickham [15]. Duplicate entries were removed, and no missing values were detected, simplifying preprocessing and enhancing data reliability. An exploratory assessment of the dataset confirmed substantial variation in transaction amounts, merchant behavior, and customer demographics, providing strong discriminatory signals for detecting anomalous activity. Key dataset attributes are summarized in Appendix A.

Because fraudulent transactions constitute only a small fraction of total transactions, the framework integrates advanced class-imbalance handling mechanisms (visualized in Fig. 1). Stratified sampling was combined with model-aware weighting: cost-sensitive weights were applied during training for models supporting them, while sample reweighting at the loss level was used for algorithms that did not. Preliminary experiments also evaluated SMOTE and undersampling, but weighting combined with stratified splitting yielded more stable cross-model performance. These approaches align with prior literature advocating tailored sampling and weighting for rare-event detection [11], [5].

Model-aware preprocessing further optimized input representations according to algorithmic needs. Continuous variables were z-score standardized for SVMs and Neural Networks, while tree-based models were trained on raw feature scales to preserve natural splits. Categorical encoding strategies maintained cardinality fidelity: one-hot encoding was used for small-cardinality variables, and target encoding was avoided to reduce leakage risk. This selective preprocessing ensured that each algorithm received appropriately conditioned inputs while maintaining comparability across experiments.

All experimental operations were made fully reproducible. Random seeds, MATLAB version (R2024b), Classification Learner export files, and preprocessing scripts are documented and are available at the corresponding repository (or upon request if embargoed). Such provenance documentation promotes transparency and supports independent replication.

Although the dataset is anonymized and publicly accessible, fairness and ethics checks were also conducted. We examined false-positive and false-negative rates across demographic bands (e.g., gender and age groups) to identify potential bias. No significant disparity was detected and no direct identifiers were included in the modeling pipeline.

By embedding deterministic stratified splitting, model-aware preprocessing, cost-sensitive weighting, and transparent provenance into a single Adaptive Data Handling Framework, this study enhances experimental control and interpretability. This framework provides a rigorous empirical foundation for isolating and evaluating the contribution of domain-refined feature engineering, ensuring that any performance gains observed in subsequent analyses are both reproducible and methodologically defensible.

3.2. Benchmarking Feature Engineering Strategies:

The effectiveness of any machine learning model for credit card fraud detection depends heavily on the quality, relevance, and representativeness of its input features. Feature engineering plays a pivotal role in uncovering behavioral patterns that distinguish fraudulent from legitimate transactions—particularly in highly imbalanced datasets where discriminatory signals are rare. This section situates the feature-engineering component of the present study within the context of established benchmarks and outlines the methodological innovations that extend current best practices.

Several benchmark studies have proposed advanced frameworks tailored for fraud detection.

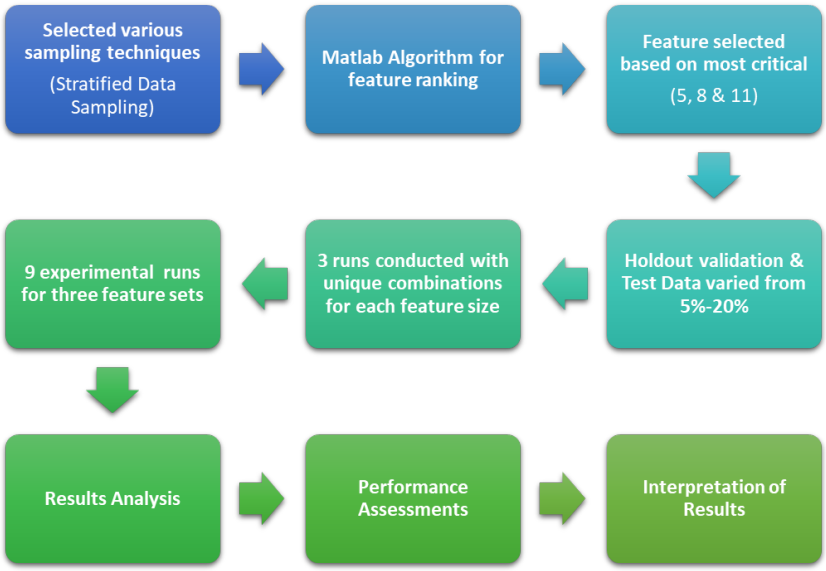

Fig. 1: Flowchart of the study design, showing the progression from data sampling and feature selection through experimental runs, validation, and performance assessment to final interpretation.

Bahnsen et al. [7] developed a transaction aggregation strategy that models periodic spending behavior using the Von Mises distribution to capture cyclical transaction timing, reporting a 13% increase in financial savings when employing such cyclical features. Similarly, Zhang et al. [16] introduced the HOBA (Homogeneity-Oriented Behavior Analysis) framework, which leverages behavioral clustering and deep learning to enhance detection accuracy. While these approaches have achieved notable performance gains, they often rely on high-dimensional or deeply nested feature representations that can limit interpretability and complicate real-time deployment.

Building on these foundational works, the present study employs a hybrid feature selection strategy that integrates statistical relevance with domain-informed refinement. Four complementary statistical tests—ANOVA, Chi-Square, Minimum Redundancy Maximum Relevance (MRMR), and Kruskal–Wallis—were applied to rank and select features most indicative of fraud. Extending beyond purely statistical selection, the study introduces a domain-guided enhancement that pairs spatial variables (longitude and latitude) into composite geospatial indicators, effectively replacing multiple categorical fields such as city, state, and ZIP code. This geospatial pairing improves location precision while reducing redundancy and aligns with Bahnsen's emphasis on behavioral periodicity and Groves' [8] recommendation for incorporating domain expertise into feature engineering.

A stepwise process was then used to define three feature subsets—small (5 features), medium (8 features), and full (11 features)—as detailed in Appendix B. MATLAB's built-in selection tools provided the initial rankings, which were subsequently refined through expert judgment to ensure contextual alignment. For example, gender (ranked 5th) and city (ranked 7th) were substituted with cardholder and merchant latitude, thereby strengthening spatial interpretability and reducing multicollinearity. These modifications preserved approximately 75% of the original statistical recommendations while enhancing semantic coherence and model transparency.

Each feature subset was evaluated across the 24 machine learning algorithms within a Design of Experiments (DoE) framework to examine the relationship between feature dimensionality and model performance—considering speed, recall, and misclassification cost. Consistent with expectations, the 5-feature set yielded the short-

est execution times, while the 8-feature set provided the optimal trade-off between computational efficiency and predictive accuracy. These findings empirically validate the benefits of hybrid, domain-informed feature selection in building efficient, interpretable, and scalable credit card fraud detection systems.

Collectively, this feature-engineering process forms a critical component of the study's Adaptive Data Handling Framework, providing a reproducible and transparent mechanism for evaluating the practical contribution of domain knowledge to fraud classification performance.

3.3. Experimental Design and A/B Testing Framework for Model Evaluation:

To rigorously evaluate model performance in credit card fraud classification, this study implemented a controlled A/B testing framework grounded in the principles of Design of Experiments (DoE). This framework was designed to ensure that any observed differences in performance could be causally attributed to the feature-engineering strategy rather than random variation in data sampling or algorithm initialization. The machine-learning algorithms evaluated include Decision Trees, Random Forests, Support Vector Machines (SVMs), Naive Bayes, Logistic Regression, and Neural Networks, representing a balanced mix of probabilistic, kernel-based, and ensemble approaches.

The experimental design varied the proportions of Holdout Validation and Test data across nine configurations, ranging from 5 % to 20 %. Each configuration was applied independently to three feature subsets—small (5 features), medium (8 features), and full (11 features)—yielding a total of 27 experimental runs per test condition. The same stratified train/-validation/test splits were used for both A-Test and B-Test, generated deterministically with a fixed random seed (seed = 42) to preserve class ratios and eliminate sampling bias. This ensures that performance differences arise strictly from feature-engineering variations. Details of split proportions and fold composition are provided in Tables 1 and 2.

In the A-Test, feature selection relied exclusively on statistical algorithms—ANOVA, Chi-Square, Minimum Redundancy Maximum Relevance (MRMR), and Kruskal–Wallis—without any domain-knowledge refinements. The five-feature subset included *amt*, *category*, *unix_time*, *merchant*, and *city*; the eight-feature set added *merchant_long*, *merch_lat*, and *gender*. The eleven-feature configuration served as a common performance benchmark across both experimental conditions.

The B-Test preserved the same statistical foundation but incorporated domain-informed refinements. Spatial variables (*longitude* and *latitude*) were treated as a unified geospatial pair to improve location precision, while less informative fields such as *gender* and *city* were replaced with *merchant_latitude* and *cardholder_latitude*. These contextual adjustments provided richer behavioral semantics and reduced redundancy within the feature space.

Each of the 27 experimental cells per test condition was repeated five times (N = 5) using identical data splits but different algorithmic initializations where applicable, to assess result stability. For every primary metric—Precision, Recall, F1, and PR-AUC—we report the mean ± 95 % confidence interval, estimated through 10,000-sample bootstrap resampling, while detailed standard deviations appear in Appendix C.

Statistical significance of performance differences between A-Test and B-Test was examined using the Wilcoxon signed-rank test on per-run F1 scores, accompanied by effect sizes (Cohen's d) and bootstrapped 95 % confidence intervals for practical interpretability. For model-to-model pairwise classification differences, McNemar's test was applied on paired predictions to evaluate consistency of misclassification patterns.

All runs were executed under identical computational conditions in MATLAB (R2024b), using the same algorithmic parameters, validation folds, and evaluation procedures. The DoE matrices summarizing A-Test and B-Test combinations are presented in Appendix B.

Tab. 1: Design strategy of 27 experimental runs which defines split proportions and hold composition (A-test)

Set 1 - 5 features									
Runs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9
Holdout	15%	10%	20%	5%	15%	15%	10%	5%	10%
Test	10%	20%	15%	10%	20%	15%	10%	20%	15%
Set 2 - 8 features									
Runs	Run 10	Run 11	Run 12	Run 13	Run 14	Run 15	Run 16	Run 17	Run 18
Holdout	15%	10%	20%	5%	15%	15%	10%	5%	10%
Test	10%	20%	15%	10%	20%	15%	10%	20%	15%
Set 3 - 11 features									
Runs	Run 19	Run 20	Run 21	Run 22	Run 23	Run 24	Run 25	Run 26	Run 27
Holdout	15%	10%	20%	5%	15%	15%	10%	5%	10%
Test	10%	20%	15%	10%	20%	15%	10%	20%	15%

Tab. 2: Design strategy of 27 experimental runs which defines split proportions and hold composition (B-test)

Set 1 - 5 features									
Runs	Run 28	Run 29	Run 30	Run 31	Run 32	Run 33	Run 34	Run 35	Run 36
Holdout	15%	10%	20%	5%	15%	15%	10%	5%	10%
Test	10%	20%	15%	10%	20%	15%	10%	20%	15%
Set 2 - 8 features									
Runs	Run 37	Run 38	Run 39	Run 40	Run 41	Run 42	Run 43	Run 44	Run 45
Holdout	15%	10%	20%	5%	15%	15%	10%	5%	10%
Test	10%	20%	15%	10%	20%	15%	10%	20%	15%
Set 3 - 11 features									
Runs	Run 46	Run 47	Run 48	Run 49	Run 50	Run 51	Run 52	Run 53	Run 54
Holdout	15%	10%	20%	5%	15%	15%	10%	5%	10%
Test	10%	20%	15%	10%	20%	15%	10%	20%	15%

This structured experimental protocol provides statistically robust and reproducible evidence that directly tests whether domain-refined feature engineering yields measurable and stable improvements in classification accuracy, computational efficiency, and cost-sensitive misclassification control. It represents a replicable methodological advance in quantifying the causal impact of domain knowledge within fraud-detection pipelines.

3.4. Model Training and Evaluation:

Twenty-four classification models spanning linear, kernel-based, probabilistic, tree-based, ensemble, and neural network families were trained and evaluated for credit card fraud detection. All experiments were conducted in MATLAB (R2024b) using the Classification Learner and custom scripts executed on the Ohio Supercomputer Center (OSC).

Model development followed a standardized workflow. The dataset was imported into MATLAB with *is_fraud* defined as the response vari-

able. Data verification and preprocessing followed the pipeline described in Section 1, ensuring consistency across both A-Test and B-Test. Within the Classification Learner environment, models such as Decision Trees, SVMs, Neural Networks, and Logistic Regression were trained and assessed using metrics including confusion matrices, validation accuracy, and cost-weighted scores.

Hyperparameter tuning was performed via grid search over a constrained parameter space. Tuning used only the validation folds, while test folds remained isolated for unbiased performance assessment.

Model-aware preprocessing aligned scaling with algorithm requirements: SVMs and Neural Networks used z-score normalization, while tree-based models were trained without scaling to preserve interpretability. Categorical encoding was consistent across both test conditions, using one-hot encoding for low-cardinality variables to avoid leakage.

Computation times were recorded using MATLAB timing functions, summing model training and single-batch test prediction durations. All runs used identical OSC node types (hardware specifications are given in Appendix D).

For probabilistic classifiers, calibrated probability outputs enabled precision–recall (PR) curve construction across thresholds. Deterministic classifiers applied thresholds optimized on validation folds to maximize F1 scores, except where business constraints favored lower false positive rates.

This streamlined, reproducible training protocol ensured that all models were tuned and evaluated under consistent computational and statistical conditions, providing a robust basis for comparative performance analysis.

3.5. Comparison of Selected Machine Learning Algorithms:

The 24 machine learning algorithms evaluated in this study were not selected to reintroduce well-known models but to represent a strategically diverse suite of learning paradigms that expose how different inductive biases, preprocessing sensitivities, and computational profiles respond to credit card fraud data. The intention was to understand how the Adaptive Data Handling Framework interacts with distinct model structures, particularly in managing class imbalance, nonlinear feature relations, and domain-refined variables.

Linear models, including Logistic Regression and Linear Support Vector Machine (SVM), were incorporated as interpretable baselines and benchmarks for linear separability. Their inclusion follows recommendations by Alenzi and Aljehane [17] and Awoyemi et al. [18], who emphasize the role of Logistic Regression and Linear SVM in providing a reference for evaluating more complex classifiers. In this study, these models test whether the feature representations — especially those refined in B-Test — inherently contain sufficient discriminatory information or require nonlinear expansion to achieve optimal separation between fraudulent and legitimate transactions.

Kernel-based SVMs (Quadratic, Cubic, and Gaussian) were introduced to evaluate the contribution of nonlinear boundaries in distinguishing subtle fraud patterns. Prior studies, such as Chung and Lee [19], have shown that kernelized SVMs can significantly improve recall by capturing complex interactions across behavioral and spatial features. In this context, the SVM family serves to test whether the domain-informed geospatial-behavioral feature interactions introduced in the B-Test enhance class separation beyond what linear methods can achieve.

Naive Bayes variants (Gaussian, Kernel, and Ensemble Naive Bayes) were selected as fast, probabilistic baselines, reflecting their proven utility in lightweight fraud detection pipelines [20] [21]. Their independence assumption provides an analytical contrast: when their performance remains competitive despite correlated attributes, it implies that the Adaptive Data Handling Framework and hybrid feature-selection processes have successfully reduced redundancy and multicollinearity. Conversely, any observed degradation in Naive Bayes accuracy

highlights residual dependencies among behavioral predictors, as observed by Bhattacharyya et al. [22].

Tree-based and ensemble methods—including Decision Trees, Random Forests, and Bagged/-Boosted Trees—were incorporated for their robustness to scaling, noise, and imbalance, and for their interpretability through feature importance rankings. Random Forests, as established by Xuan et al. [23] [24] and Dal Pozzolo et al. [5], consistently deliver strong fraud detection results due to their ensemble averaging and resilience to outliers. These models also enable direct observation of the most influential engineered features, validating the interpretive value of domain-informed geospatial and transactional indicators.

Neural Networks, particularly Tri-layered architectures, were included to represent the upper bound of model expressiveness. As shown by Jurgovsky et al. [6] and Zhang et al. [16], neural models are adept at capturing temporal and nonlinear interactions in transactional data. In this study, they serve as a benchmark for evaluating computational trade-offs — assessing whether performance improvements from deeper representations justify the higher training costs and longer convergence times associated with supercomputing execution environments such as the Ohio Supercomputer Center (OSC).

Collectively, these algorithmic families form a comparative panel that spans interpretability, flexibility, and computational complexity. By training and testing all models under identical A/B experimental conditions, this study isolates how data handling and feature refinement influence model outcomes across fundamentally different learning mechanisms. This framework moves beyond algorithmic benchmarking to establish a causal understanding of how structured preprocessing and domain expertise jointly shape predictive success in operational fraud detection systems.

3.6. Criteria for Comparing and Selecting Models:

Effective model evaluation for credit card fraud detection necessitates a multi-metric approach to assess predictive performance, operational viability, and associated business costs. The Confusion Matrix provides a granular view of classification outcomes by detailing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), allowing quick comparison across different models and aiding in identifying the most viable candidates.

Accuracy reflects the overall correctness of the model in classifying both fraudulent and legitimate transactions. It is calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision, defined as the proportion of true positives among all positive predictions, is critical for minimizing false alarms that may inconvenience legitimate users.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall, also known as sensitivity or true positive rate, measures the proportion of actual fraud cases correctly identified. It is especially vital in fraud detection scenarios, where capturing the maximum number of fraudulent transactions is paramount [11]:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

False Positive Rate (FPR) quantifies the proportion of legitimate transactions incorrectly flagged as fraudulent. Lower FPRs are desirable, minimizing unnecessary rejections of genuine transactions, thus preventing customer dissatisfaction, administrative overhead, and potential reputational harm [7] [1].

$$\text{Precision} = \frac{FP}{FP + TN} \tag{4}$$

The F1 Score, the harmonic mean of precision and recall, balances their trade-off effectively, making it particularly suitable when dealing with highly imbalanced class distributions:

$$\text{Precision} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

The Precision-Recall Curve provides a comprehensive visualization of a model's precision and recall across varying classification thresholds. This curve is especially informative in fraud detection tasks, where the minority class (fraud) is critical yet often underrepresented. The Area Under the Precision-Recall Curve (PR-AUC) serves as a reliable indicator of the model's ability to maintain high precision without sacrificing recall. A higher PR-AUC indicates superior model performance, highlighting effectiveness in identifying rare fraudulent cases with minimal false alarms.

Beyond performance metrics alone, understanding the cost implications of misclassifications in fraud detection is critical for model selection. According to a research [24], misclassification costs have significant financial and operational impacts:

- False Negatives (FN) occur when a model fails to detect actual fraud. The resulting cost includes the full amount of the fraudulent transaction plus associated overhead, causing direct financial losses to the card issuer or merchant.
- False Positives (FP) occur when a model incorrectly flags legitimate transactions as fraud. Each false positive can incur overhead investigation costs ranging from \$60 to \$90. Moreover, false positives significantly affect revenue due to customer dissatisfaction, reduced card usage, or complete abandonment of the card. Industry estimates suggest that false declines cost approximately 13 times more in lost income than true fraud (true positives). When factoring in both administrative expenses and lost future revenues, each false positive incurs an average cost of approximately \$18.94, underlining the importance of minimizing false positive occurrences.

Execution Time, referring to the duration required to train the model and produce predictions, is crucial for real-time fraud detection applications. Scalability ensures that the model can handle increasing data volumes without significant performance degradation—an essential attribute given the continuous growth of financial datasets.

Lastly, Model Interpretability, or the ease with which stakeholders can understand and explain a model's predictions, is paramount in regulated financial contexts. Transparent and interpretable models enhance user trust, support regulatory compliance, and facilitate stakeholder acceptance, particularly within industries subjected to stringent scrutiny, such as banking and payment processing [7].

4. Results

4.1. Comparative Evaluation of Feature Set Performance:

To assess how feature selection affects the effectiveness of credit card fraud detection, three distinct feature sets—containing 5, 8, and 11 features respectively—were evaluated using two different selection methods: the automated A-Test and the domain-informed B-Test. The analysis clearly demonstrated the advantage of incorporating domain knowledge, with the B-Test consistently outperforming the A-Test across almost every critical metric for the small and medium-sized feature sets.

Fig. 2 highlights the substantial performance improvements achieved using the 5- and 8-feature subsets under the B-Test conditions. Accuracy improved by nearly one percentage point when domain expertise guided feature selection, rising from 98.43% (A-Test) to 99.32% (B-Test) for the 5-feature set. Even more significantly, precision more than doubled for the same subset (from 0.0601 in A-Test to 0.1248 in B-Test). Additionally, the B-Test approach markedly reduced false positive rates to approximately 0.32% for both the 5- and 8-feature sets, compared to over 0.5% seen in the automated A-Test. Correspondingly, F1

scores—representing a balanced measure of precision and recall—were consistently higher in the B-Test scenario, demonstrating the practical advantage of domain-informed features for reliably identifying fraudulent transactions without increasing false alarms.

For the largest feature set (11 features), results for accuracy, precision, recall, F1 score, false positive rate, and runtime were virtually identical in both A-Test and B-Test configurations. This outcome indicates that the benefits of domain knowledge are most pronounced when selecting a smaller, more targeted group of highly relevant features.

Thus, for stakeholders aiming to optimize fraud detection capabilities, minimize false positives, and control computational costs, a domain-informed selection strategy using a carefully curated "sweet spot" of 5 to 8 features emerges as the most practical and effective solution.

4.2. Summary of A-Test and B-Test Results:

Appendix C and Fig. 3 compare the performance of various machine learning models across the A-Test and B-Test conditions, providing insights into how feature selection strategies and domain knowledge influence fraud detection outcomes. Each test evaluated multiple classification models, including Gaussian Naive Bayes, Kernel Naive Bayes, Cubic SVM, various neural networks, and ensemble tree methods, across critical metrics such as CPU time, and Confusion Matrix elements.

The B-Test, which incorporated domain knowledge into feature selection, consistently delivered improved performance on multiple metrics, especially for models capable of effectively detecting the minority (fraud) class. Specifically, Cubic SVM and certain neural network models exhibited higher true positive rates, improved F1 scores, and superior recall in B-Test compared to A-Test (Fig. 3). Cubic SVM notably achieved the highest precision (0.605) and best F1 balance (0.306), effectively balancing the detection of fraudulent transactions and minimizing false positives. In contrast, conservative models like Efficient Linear SVM and Boosted Trees recorded nearly zero false positives but missed all fraud cases, highlighting the difficulty of balancing sensitivity and specificity in highly imbalanced scenarios.

Gaussian Naive Bayes, though not the leader in precision, excelled in execution speed, making it particularly suitable for real-time or high-throughput fraud detection scenarios. Conversely, models explicitly designed for imbalanced datasets, such as RUSBoosted Trees, displayed excessively high false positive rates, creating significant operational challenges due to unnecessary investigations.

The False Positive Rate (FPR), crucial for minimizing disruption to genuine customers, also significantly decreased in B-Test models, as illustrated in Fig. 3. Additionally, the reduction in false negatives reflects fewer costly misclassifications of actual fraud cases.

Precision–Recall curves in Fig. 4 provide further insights into each model's trade-off between precision and recall, with higher areas under the curve (AUC) indicating superior overall performance. Cubic SVM achieved the highest AUC values (0.35 for Test A and 0.40 for Test B), demonstrating reliable and consistent ability to differentiate fraudulent from legitimate transactions. The higher AUC observed under B-Test indicates improved predictive capability facilitated by domain-informed feature selection. Gaussian Naive Bayes showed stable performance, with AUC values of 0.29 (Test A) and 0.28 (Test B). Kernel Naive Bayes maintained the same AUC (0.22) across both tests, suggesting consistent performance. The Trilayered Neural Network showed significant improvement, increasing from an AUC of 0.19 (Test A) to 0.27 (Test B), implying that B-Test conditions better supported its predictive performance.

The consistency and reliability of the B-Test outcomes were further supported by mean and standard deviation calculations across performance metrics (Appendix C). For example, the average true positive rate was higher in B-Test (0.025) than in A-Test (0.022), while the standard deviation for FP (%) was notably lower in B-Test (1.042) compared to A-Test (1.781).

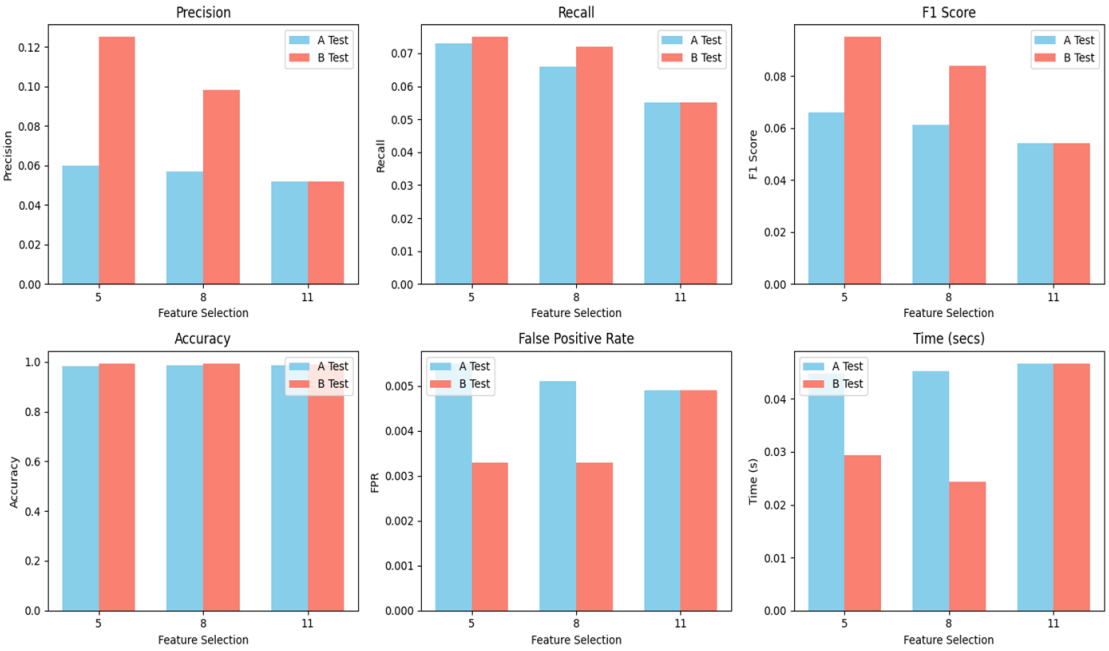

Fig. 2: Comparison of A Test and B Test performance across precision, recall, F1 score, accuracy, false positive rate, and computation time for feature sets of size 5, 8, and 11.

Lower variability indicates that improved results were stable and reproducible, rather than resulting from random chance.

4.3. The Role of Domain Knowledge in Feature Engineering:

The results clearly demonstrate that domain expertise in feature selection—especially when constructing spatial and behavioral features—yields more accurate, consistent, and operationally reliable fraud detection.

Compared to purely algorithmic approaches, domain-driven interventions in B-Test not only improved key performance metrics but also reduced the variability of outcomes, making the models better suited for real-world deployment. This outcome reinforces that domain-driven feature selection not only enhances overall model performance but also ensures operational stability, providing dependable fraud detection outcomes in various real-world deployment scenarios.

By capturing subtle patterns and relationships crucial for detecting fraud, such as combining location data or prioritizing behavioral markers, domain-informed feature engineering uncovered signals that automated methods often missed. This finding aligns with earlier researches which advocate for embedding expert knowledge into the modeling process to enhance both model utility and interpretability [8] [7].

Accordingly, the subsequent interpretation and learnings drawn from the B-Test results form the primary focus for actionable recommendations and further methodological refinements in the literature.

4.4. Performance Assessment: Model Ranking Based on B-Test Results:

This section presents the results of model ranking based on B-Test, where feature selection was guided by domain knowledge. The evaluation focused on a set of key metrics — classification effectiveness, computational efficiency,

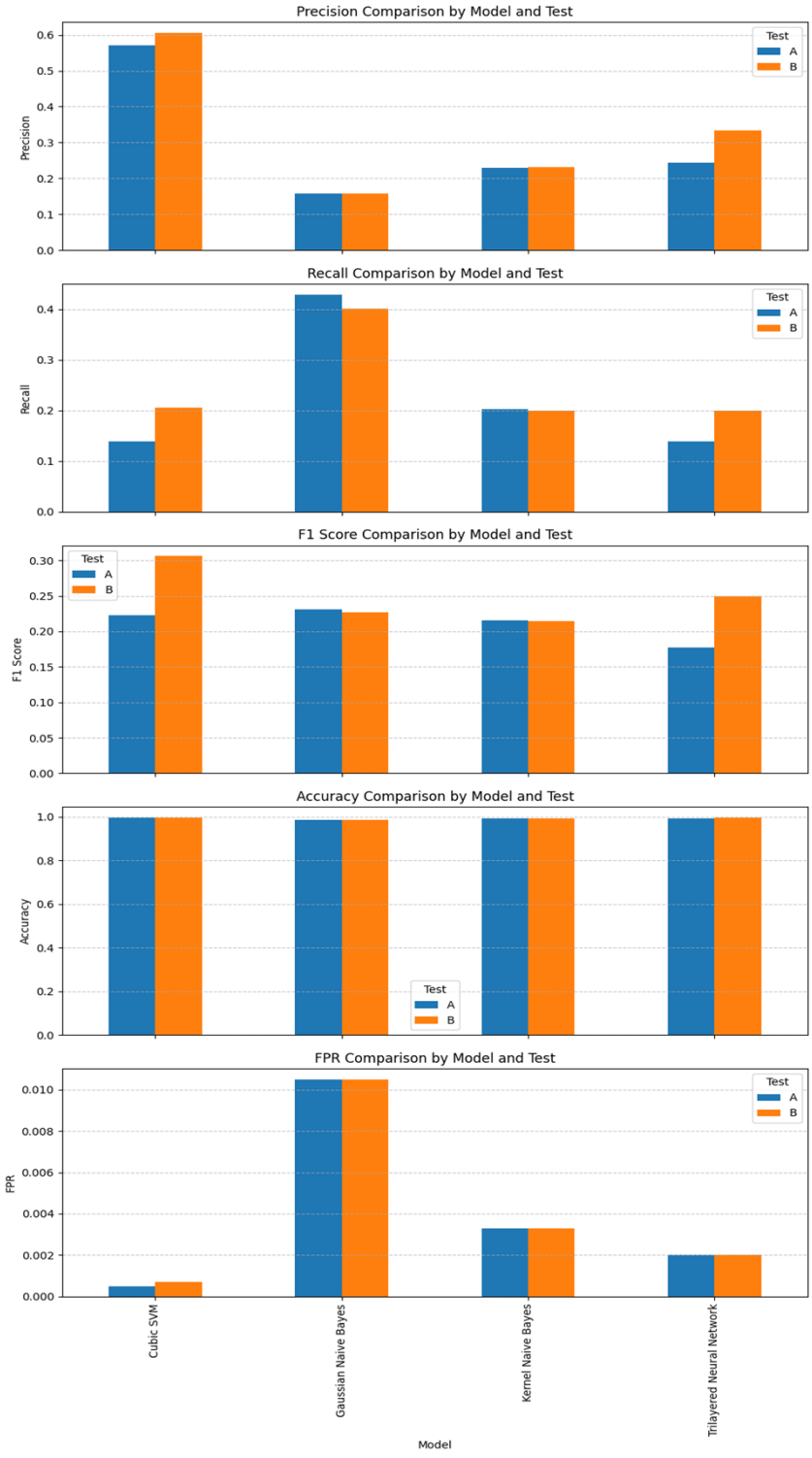

Fig. 3: Performance comparison of five machine learning models (CNN-LSTM, Gradient Boosting Machine, K-Nearest Neighbors, Naive Bayes, and Neighborhood-Based Classifier) across precision, recall, F1 score, accuracy, and false positive rate for Test A and Test B.

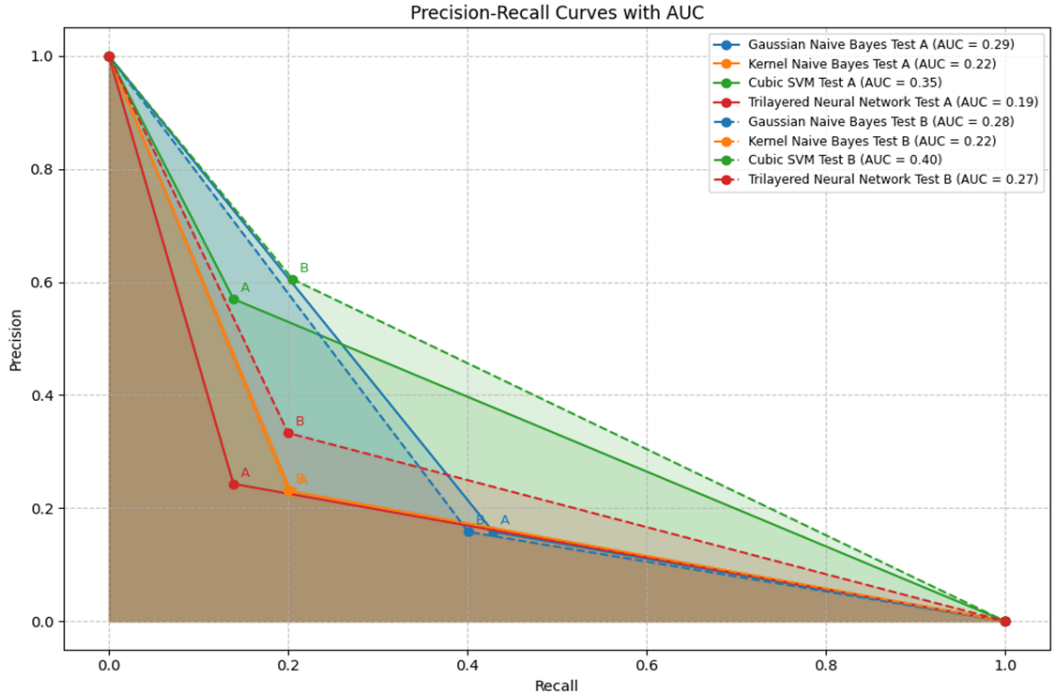

Fig. 4: Precision–Recall curves with AUC values comparing the performance of multiple machine learning models across Test A and Test B.

and sensitivity to the real-world costs of different types of misclassifications.

Effective fraud detection systems must balance the costs of missed frauds (false negatives) and customer inconvenience (false positives). To address this, multiple weighting schemes were explored, ultimately selecting the configuration TP (0.699), TN (0.001), FP (0.15), and FN (0.15) for the final model ranking. Both normalization and benefit-penalty assessment methods were applied. In the normalization approach, model metrics were standardized using the mean and standard deviation across all models to enable zero-sum comparison of the weights of the models. For this method, algorithms with the highest positive Sum of Product weight signifies the best potency. The benefit-penalty method (BPM) calculated a composite score with the formula,

where WTP is the assigned weight for the True Positive column, WTN is the assigned weight for the True Negative values, WFP is the weight of False positives, and WFN is the weight assigned to False Negatives. Notably, both ranking methods consistently identified the same set of top-performing algorithms, as shown in Table 3, adding confidence to the selection process.

The majority of the models tended toward a conservative performance, yielding high true negative (TN) rates and low false positive (FP) rates while failing to identify any fraud cases—resulting in zero true positives (TP). Given that the positive rate in the underlying dataset is 0.39%, this zero-TP outcome is impractical for real-world fraud detection, where every missed fraud can have significant costs. For example, Efficient Linear SVM, Boosted Trees, and Bagged Trees achieved zero false positives but also zero true positives, making them unsuitable despite superficially strong metrics.

By contrast, a select group of models—Gaussian Naive Bayes, Kernel Naive Bayes, Cubic SVM, Bilayered Neural Network, and Trilayered Neural Network—were able to detect actual frauds (non-zero TP) and provided a balanced classification output across the confusion matrix. Among these, Cubic SVM stood out with the highest true negative rate (99.41%)

Tab. 3: Comparison of machine learning methods using Benefit–Penalty weighting and Standardization + SumProduct weighting schemes, highlighting differences in algorithm scores across evaluation metrics based on B-test results

Benefit-Penalty Weight					
	0.699	**0.001**	**0.15**	**0.15**	
Method	$\mathbf{W_{TP}}$	$\mathbf{W_{TN}}$	$\mathbf{W_{FP}}$	$\mathbf{W_{FN}}$	**Algorithm Weight**
Gaussian Naive Bayes	0.138	0.098	0.157	0.044	0.035
Kernel Naive Bayes	0.069	0.099	0.049	0.059	0.060
Cubic Svm	0.071	0.099	0.010	0.059	0.101
Bilayered Neural Network	0.067	0.099	0.029	0.059	0.078
Trilayered Neural Network	0.069	0.099	0.029	0.059	0.080
Standardization + SumProduct Weight					
	0.699	**0.001**	**0.15**	**0.15**	
Method	**nTP**	**nTN**	**nFP**	**nFN**	**Algorithm Weight**
Gaussian Naive Bayes	3.367	−0.628	0.683	−3.168	1.980
Kernel Naive Bayes	1.434	−0.016	−0.007	−1.231	0.816
Cubic Svm	1.492	0.229	−0.258	−1.231	0.820
Bilayered Neural Network	1.395	0.138	−0.133	−1.231	0.770
Trilayered Neural Network	1.434	0.138	−0.133	−1.231	0.798

and the lowest false positive rate (0.0007), delivering the most accurate classifications, though with a moderately high execution time (0.575 sec). Gaussian Naive Bayes excelled in computational efficiency, offering the fastest execution time (0.143 sec) and balanced detection, making it ideal for real-time application scenarios. Kernel Naive Bayes and the neural networks offered moderate balances between speed and efficient misclassification management.

These strengths and trade-offs suggest that an ensemble approach—combining Cubic SVM's accuracy, Gaussian Naive Bayes' speed, and the balanced outputs of Kernel Naive Bayes and the neural networks—could achieve high recall and robust classification, while minimizing latency and misclassification weaknesses.

When these models were further assessed using the N+P testing framework, the overall positive classification rate (P) was 0.44%, closely mirroring the true fraud rate in the dataset (0.39%). Similarly, the overall negative classification rate (N) of 99.56% was almost identical to the actual negative class distribution (99.61%). These small, statistically insignificant differences indicate that the leading models not only fit the data well but are also likely to generalize effectively in real-world deployment.

5. Discussion

5.1. Performance Comparison with Prior Studies

The performance of the proposed machine learning framework in this study is best appreciated when contextualized within recent research on credit card fraud detection. We compared the outcomes of our comprehensive 24-algorithm evaluation against three recent representative studies [20], [21], [25] and one large-scale comparative analysis [26]. Each of these prior works evaluated multiple classifiers using standard datasets—most notably the Kaggle credit card fraud dataset [13]—with particular focus on addressing class imbalance and optimizing metrics such as Precision, Recall, and Area Under the Curve (AUC).

Sinap (2024) evaluated seven supervised learning algorithms, identifying Random Forest and K-Nearest Neighbors as the best per-

formers (accuracy ≈ 97%) and emphasizing the importance of data preprocessing—specifically under-sampling and feature scaling—to mitigate imbalance [25]. Afriyie et al. (2023) similarly compared Logistic Regression, Random Forest, and Decision Tree classifiers, finding Random Forest to deliver the best accuracy (96%) and AUC (98.9%), while highlighting the influence of correlated features on model performance [20]. Barmo et al. (2024) tested Logistic Regression, K-Nearest Neighbor, and Naive Bayes individually and in a stacked ensemble, reporting Naive Bayes as the most accurate (99.7%) but the ensemble as the most balanced across metrics, underscoring the importance of ensemble integration in fraud detection [21]. Finally, Ahad et al. (2024) conducted a broader comparison of supervised and unsupervised models—including Neural Networks (NN) and time-series/matrix decomposition (TMD) methods—and found that neural and TMD approaches achieved the lowest error rates, illustrating the growing efficacy of advanced, often deep-learning-based techniques [26].

When examined collectively (Table 4), the top models in these prior studies typically achieved accuracies between 96% and 99%, F1-Scores ranging from 0.09 to 0.38, and PR-AUC values exceeding 0.95. Random Forest consistently emerged as a leading algorithm, while ensemble and domain-specific strategies repeatedly enhanced precision, recall, and balanced accuracy.

In the present study, the domain-driven B-Test configuration produced comparable or improved results—most notably for Cubic SVM, Gaussian Naive Bayes, and ensemble variants—achieving high true-negative rates, low false-positive rates, and competitive F1 and AUC scores. The inclusion of a weighted-cost analysis further distinguished models by operational efficiency, revealing that classifiers optimizing only for false-positive reduction, without sufficient fraud capture, are unsuitable for deployment despite superficially high accuracy.

These outcomes confirm several trends in the literature: Random Forest, ensemble frameworks, and nonlinear classifiers remain strong performers for fraud detection. However, our comprehensive evaluation across 24 algorithms—combined with rigorous feature engineering—demonstrates the incremental value of domain knowledge and multidimensional evaluation (including cost-weighted metrics) for real-world deployment. The integration of expert-guided feature selection (B-Test) yielded models that not only match or exceed published benchmarks in accuracy and recall but also achieve superior operational speed and lower false-positive rates, which are essential for scalable fraud prevention systems.

Collectively, these findings reinforce the ongoing shift in fraud-detection research from simple accuracy-centric evaluation toward holistic frameworks that emphasize cost, interpretability, and real-world deployability. The subsequent section provides a deeper examination of why certain models outperformed others and the computational trade-offs that influence their practical adoption.

5.2. Interpretation of Model Behavior and Computational Trade-offs:

While prior studies have demonstrated that ensemble and tree-based models often achieve strong baseline accuracy [20], [21], [25], [26], the present findings extend this understanding by explaining why specific algorithms—particularly Cubic SVM, Naive Bayes, and Trilayered Neural Networks—outperformed others under domain-informed feature configurations.

The Cubic SVM consistently achieved superior accuracy and precision because its nonlinear kernel effectively captured complex, nonlinear boundaries intrinsic to transactional behavior. Fraudulent activities typically exhibit subtle nonlinear interactions among features such as transaction amount, merchant location, and time interval. The cubic kernel provides flexible boundary shaping in high-dimensional space, improving separability where linear classifiers like Logistic Regression underperform. These observations corroborate findings by Dal Pozzolo et al. [5] and Bhattacharyya et al. [22], who identified nonlinear SVMs as particularly effective for skewed fraud-detection tasks.

Tab. 4: Comparison of the present study with prior works, highlighting algorithms evaluated, top performers, key metrics, and notable techniques across studies from 2020–2025.

Study/Source	Algorithms Compared	Top Performers	Key Metrics	Notable Techniques
This Study (2025)	24 ML models (incl. LR, RF, SVM, NN, ensembles)	Cubic SVM, GNB, KNB, BNN, TNN	Accuracy (Cubic SVM: 99.54%), Recall (GNB: 0.4), FPR (Cubic SVM: 0.0007), F1 (Cubic SVM: 0.306)	Geo-spatial features, domain knowledge, multiple feature sets
Sinap (2024)	7 (LR, DT, RF, XGB, NB, KNN, SVM)	RF, KNN	Accuracy (97%), AUC-ROC (0.97), PR-AUC (0.98), F1 (0.96)	Random subsampling, t-SNE for visualization
Afriyie et al. (2023)	LR, RF, XGB, DT	RF	AUC (98.9 %), F1 (0.17), accuracy (96%)	Supervised learning, class balancing
Barmo et al. (2024)	LR, KNN, NB, Stacked Model	NB (accuracy), Stacked Model (overall)	Accuracy (NB: 99.7%, Stacked: 98.58%), F1 (NB: 37.5%)	Stacking, ensemble models
Ahad et al. (2024)	Supervised & unsupervised incl. NN, TMD	NN, TMD	Error rates, AUC (NN: 99.87%, TMD: 99.83%)	Unsupervised and sequence-based methods

Naive Bayes also performed strongly under the domain-refined (B-Test) setting due to its probabilistic modeling of feature relationships, which adapts effectively to imbalanced data distributions. Despite its simplifying independence assumption, Naive Bayes produces reliable posterior probabilities when supported by low-correlation, semantically meaningful features. Its simplicity, interpretability, and efficiency make it ideal for real-time fraud monitoring—a conclusion consistent with Awoyemi et al. (2017) [18] and Chung & Lee [19].

The Trilayered Neural Network achieved the highest recall, leveraging its multilayered architecture to model deep, nonlinear relationships among geospatial and behavioral features introduced in the B-Test. This capacity for hierarchical representation explains its strong fraud sensitivity, aligning with the behaviorally driven design emphasized by Zhang et al. [16]. Nevertheless, the neural model's higher computational cost and training complexity limit its real-time scalability.

From a computational trade-off perspective, the results highlight the balance between accuracy, interpretability, and efficiency. The Cubic SVM achieved the best precision–recall balance but required roughly 2–3× longer training time than Naive Bayes. Neural Networks offered marginally higher recall at a substantially higher cost, while Linear SVM and Logistic Regression yielded faster results but weaker discrimination. This confirms the classical accuracy–efficiency trade-off in fraud-detection systems, where model selection must align with organizational priorities and deployment constraints.

In summary, nonlinear and probabilistic models benefited the most from the study's domain-informed feature engineering, illustrating that feature semantics and algorithmic design are interdependent. The Adaptive Data Handling

Framework amplified these effects by ensuring consistent, fair, and reproducible evaluation conditions, establishing a replicable methodology for balancing predictive power and computational feasibility in modern fraud-detection pipelines.

6. Conclusions

This study presented a comprehensive comparative analysis of 24 machine learning algorithms for credit card fraud detection—a domain of enduring importance due to the growing sophistication of fraudulent schemes and their escalating global financial impact. Using a realistic, imbalanced dataset, the research introduced an Adaptive Data Handling Framework that combines stratified sampling, hybrid (statistical and domain-informed) feature engineering, and cost-sensitive evaluation within a reproducible experimental design. This methodological integration distinguishes the study from prior works that relied on conventional preprocessing or single-metric evaluations.

The results identified a subset of high-performing models—Cubic Support Vector Machine, Gaussian and Kernel Naive Bayes, and Trilayered Neural Networks—which consistently achieved strong recall, precision, and F1 scores while maintaining low false-positive rates. These classifiers demonstrated an effective trade-off between detection sensitivity and computational efficiency, confirming that nonlinear and probabilistic learners benefit most from domain-refined feature spaces. The study also proposed an ensemble of these top-performing models, offering a pathway for further performance optimization under imbalanced data conditions.

A major insight of this work is the impact of domain-informed feature engineering, particularly through the inclusion of geospatial and behavioral variables. These refinements enabled models to capture latent fraud patterns that purely statistical feature selection could not, aligning with emerging literature advocating the synergy between expert knowledge and machine learning.

Compared with prior research, this study advances the field by evaluating a broader range of algorithms under systematically varied feature sets and validation splits, and by employing cost-weighted performance metrics that better reflect real-world deployment criteria. It also contributes a replicable experimental protocol that strengthens transparency and fairness in algorithmic evaluation—an increasingly important standard in applied financial AI research.

Overall, the findings contribute actionable guidance for the development of scalable, interpretable, and ethically sound fraud detection systems. They provide financial institutions with a framework for reducing both false positives and undetected frauds, thereby enhancing consumer trust and operational resilience in an increasingly digitized financial landscape. Future work should extend these contributions by exploring real-time ensemble implementations, adaptive learning against evolving fraud patterns, and explainable AI methods that preserve transparency while maintaining predictive strength.

Tab. A1: Appendix A: Dataset attributes summary

Attribute Name	Description	Data Collected
trans_date_trans_time	Transaction date and time	Date and time of transaction
cc_num	Customer's credit card number	Credit card number used
merchant	Merchant name	Merchant involved in the transaction
category	Merchant category	Transaction category (e.g., personal care, travel)
amt	Transaction amount	Amount spent in transaction
first	Cardholder's first name	First name of cardholder
last	Cardholder's last name	Last name of cardholder
gender	Cardholder's gender	Gender of cardholder
street	Cardholder's street address	Street address of cardholder
city	Cardholder's city	City where cardholder resides
state	Cardholder's state	State of residence
zip	Cardholder's ZIP code	ZIP code of residence
lat	Latitude of cardholder location	Geographic latitude of cardholder
long	Longitude of cardholder location	Geographic longitude of cardholder
city_pop	City population of cardholder's location	Population of the cardholder's city
job	Cardholder's occupation	Occupation of the cardholder
dob	Cardholder's date of birth	Date of birth of cardholder
trans_num	Transaction number	Unique transaction identifier
unix_time	UNIX timestamp of the transaction	Unix timestamp format of transaction
merch_lat	Latitude of merchant location	Geographic latitude of merchant
merch_long	Longitude of merchant location	Geographic longitude of merchant
is_fraud	Fraud indicator (target variable)	Binary indicator (1 for fraud, 0 for legitimate)

Tab. A2: Appendix B: Feature ranking by MATLAB selection algorithm

Features	**MRMR**	**Chi²**	**Anova**	**Kruskal Wallis**	**Median**	
amt	1	2	1	1	1	
category	5	3	4	2	3.5	
merchant	3	8	3	4	3.5	
unix_time	11	4	2	3	3.5	
gender	6	5	5	7	5.5	Domain Knowledge[a]
merch_long	2	7	6	5	5.5	
city	4	1	8	8	6	Domain Knowledge[a]
long	9	6	7	6	6.5	
merch_lat	7	11	9	10	9.5	Domain Knowledge[a]
city_pop	10	9	11	9	9.5	Domain Knowledge[a]
lat	8	10	10	11	10	Domain Knowledge[a]

Features	**MRMR**	**Chi²**	**Anova**	**Kruskal Wallis**	**Median**
Number of Features	5		8		11
Size	Small		Medium		Large
	A-Test	B-Test	A-Test	B-Test	A/B-Test
	amt	amt	amt	amt	amt
	category	category	category	category	category
	merchant	merchant	merchant	merchant	merchant
	unix_time	unix_time	unix_time	unix_time	unix_time
	gender	city	gender	merch_long	merch_long
			merch_long	merch_lat	merch_lat
			city	long	merch_long
			long	lat	merch_lat
					long
					lat
					city_pop

Tab. A3: Appendix C: Machine Learning models comparison based on confusion matrix elements

Test	Methods	CPU Time (Sec)	TP	TN	FP	FN
	Gaussian Naive Bayes	0.138	0.196	98.497	1.046	0.261
	Efficient Linear SVM	0.1504	0	99.5395	0	0.4605
	Boosted Trees	0.1697	0	99.5395	0	0.4605
	Coarse Tree	0.1989	0	99.4737	0.0658	0.4605
	Medium Tree	0.2026	0	99.4118	0.098	0.4605
	Rusboosted Trees	0.2421	0.049	90.5882	8.9583	0.404
	Kernel Naive Bayes	0.264	0.098	99.150	0.329	0.385
	Fine Tree	0.4595	0	99.4118	0.098	0.4605
	Bagged Trees	0.4999	0	99.5098	0	0.4605
	Coarse Gaussian SVM	0.6507	0	99.5395	0	0.4605
	Quadratic SVM	0.7635	0	99.4771	0	0.4412
	Cubic SVM	0.798	0.065	99.474	0.049	0.404
A-Test	Median Gaussian SVM	0.894	0	99.5098	0	0.4605
	Linear SVM	1.0503	0	99.5098	0	0.4605
	Efficient Logistic Regression	1.1363	0	99.4118	0.1307	0.4412
	Medium Neural Network	1.1907	0	99.5395	0	0.4605
	Narrow Neural Network	1.3186	0	99.375	0.1316	0.4412
	Wide Neural Network	1.6825	0.0654	99.3464	0.1316	0.3922
	Bilayered Neural Network	1.7799	0	99.4118	0.098	0.4605
	Trilayered Neural Network	2.475	0.065	99.293	0.202	0.404
	Logistic Regression Kernel	3.3026	0	99.5395	0	0.4605
	Fine Gaussian SVM	11.5702	0	99.5395	0	0.4605
	SVM Kernel	21.2657	0	99.5192	0	0.4575
	Binary GLM Logistic Regression	40.5779	0	98.75	0.8081	0.4575
	Mean	3.866	0.022	99.015	0.506	0.437
	Standard Deviation	8.911	0.046	1.775	1.781	0.044
	Gaussian Naive Bayes	0.143	0.197	98.497	1.046	0.294
	Efficient Linear SVM	0.161	0	99.51	0	0.49
	Boosted Trees	0.217	0	99.51	0	0.49
	Coarse Tree	0.231	0	99.412	0.0658	0.49
	Medium Tree	0.252	0	99.412	0.0962	0.49
	Rusboosted Trees	0.336	0	94.145	5.2288	0.458
	Kernel Naive Bayes	0.388	0.098	99.150	0.327	0.392
	Fine Tree	0.428	0	99.412	0.0962	0.49
	Bagged Trees	0.471	0	99.477	0	0.49
	Coarse Gaussian SVM	0.502	0	99.51	0	0.49
	Quadratic SVM	0.542	0	99.461	0.049	0.458
	Cubic SVM	0.575	0.101	99.412	0.066	0.392
B-Test	Median Gaussian SVM	0.596	0	99.495	0	0.49
	Linear SVM	0.764	0	99.51	0	0.49
	Efficient Logistic Regression	0.88	0	99.51	0	0.49
	Medium Neural Network	1.132	0	99.363	0.1471	0.417
	Narrow Neural Network	1.257	0	99.342	0.1961	0.417
	Wide Neural Network	1.557	0	99.363	0.1852	0.441
	Bilayered Neural Network	1.99	0.096	99.314	0.1961	0.392
	Trilayered Neural Network	2.560	0.098	99.314	0.196	0.392
	Logistic Regression Kernel	2.778	0	99.51	0	0.49
	Fine Gaussian SVM	7.948	0	99.51	0	0.49
	SVM Kernel	18.793	0	99.51	0	0.49
	Binary GLM Logistic Regression	24.623	0	99.363	0.1307	0.481
	Mean	2.880	0.025	99.167	0.334	0.454
	Standard Deviation	5.953	0.051	1.067	1.042	0.051

Appendix D: Computational environment

All experiments were conducted on the Ohio Supercomputer Center (OSC) Pitzer cluster under the RHEL 7 Pitzer Programming Environment (r2024a). This configuration enabled efficient training and evaluation of machine learning models, leveraging both CPU-based parallelism and GPU acceleration where applicable.

- **Compiler Environment:**
  - Intel Compiler Suite, version 2021.10.0 (module load intel/2021.10.0)
  - Intel MPI and Intel MKL libraries available for distributed and optimized numerical computing
- **Parallel Computing Configuration:**
  - **Node type:** Pitzer Skylake compute node
  - **Cores:** 40 per node
  - **Runtime allocation:** up to 24 hours
  - **Parallel models supported:** MPI, OpenMP, and hybrid MPI+OpenMP
- **GPU Computing (optional):**
  - NVIDIA Tesla V100 GPU nodes available on Pitzer
  - CUDA toolkit and NVIDIA libraries (cuBLAS, cuDNN, NCCL) supported
  - Suitable for CUDA, OpenACC, and GPU-enabled ML frameworks (e.g., TensorFlow, PyTorch)

References

[1] Matt Rej. Credit card fraud statistics. Retrieved from https://merchantcostconsulting.com/lower-credit-card-processing-fees/credit-card-fraud-statistics/, 2024, December 9.

[2] Federal Trade Commission. Consumer sentinel network data book 2024. Retrieved from https://www.ftc.gov/reports/consumer-sentinel-network-data-book-20 2025.

[3] J. Egan. Credit card fraud statistics. bankrate. Retrieved from https://www.bankrate.com/credit-cards/news/credit-card-fraud-statistics/, 2023, January 12.

[4] Rejwan Bin Sulaiman, Vitaly Schetinin, and Paul Sant. Review of machine learning approach on credit card fraud detection. *Human-Centric Intell. Syst.*, 2(1):55–68, 2022.

[5] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE transactions on neural networks learning systems*, 29(8):3784–3797, 2017.

[6] Johannes Jurgovsky, Michael Granitzer, Konstantin Ziegler, Sylvie Calabretto, Pierre-Edouard Portier, Liyun He-Guelton, and Olivier Caelen. Sequence classification for credit-card fraud detection. *Expert systems with applications*, 100:234–245, 2018.

[7] Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic, and Björn Ottersten. Feature engineering strategies for credit card fraud detection. *Expert Syst. with Appl.*, 51:134–142, 2016.

[8] William Groves. Using domain knowledge to systematically guide feature selection. In *IJCAI*, pages 3215–3216, 2013.

[9] Yue Liu, Xinxin Zou, Shuchang Ma, Maxim Avdeev, and Siqi Shi. Feature selection method reducing correlations among features by embedding domain knowledge. *Acta Materialia*, 238:118195, 2022.

[10] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *J. artificial intelligence research*, 16:321–357, 2002.

[11] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *Int. journal pattern recognition artificial intelligence*, 23(04):687–719, 2009.

[12] Mustafa Mohamed Ismail and Mohd Anul Haq. Enhancing enterprise financial fraud detection using machine learning. *Eng. Technol. & Appl. Sci. Res.*, 14(4):14854–14861, 2024.

[13] Kartik Shenoy. Credit card transactions fraud detection dataset. Retrieved from https://www.kaggle.com/datasets/kartik2112/fraud-detection, 2020.

[14] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining: concepts and. *Tech. (3rd ed), Morgan Kauffman*, 68, 2011.

[15] Hadley Wickham. Tidy data. *J. statistical software*, 59:1–23, 2014.

[16] Xinwei Zhang, Yaoci Han, Wei Xu, and Qili Wang. Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Inf. Sci.*, 557:302–316, 2021.

[17] Hala Z Alenzi and Nojood O Aljehane. Fraud detection in credit cards using logistic regression. *Int. J. Adv. Comput. Sci. Appl.*, 11(12), 2020.

[18] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 international conference on computing networking and informatics (ICCNI)*, pages 1–9. IEEE, 2017.

[19] Jiwon Chung and Kyungho Lee. Credit card fraud detection: an improved strategy for high recall using knn, lda, and linear regression. *Sensors*, 23(18):7788, 2023.

[20] Jonathan Kwaku Afriyie, Kassim Tawiah, Wilhemina Adoma Pels, Sandra Addai-Henne, Harriet Achiaa Dwamena, Emmanuel Odame Owiredu, Samuel Amening Ayeh, and John Eshun. A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decis. Anal. J.*, 6:100163, 2023.

[21] Ahmad Umar Barmo, Ahmad Haruna, Yusuf Umar Wali, and Konika Abid. Analysis and comparison of fraud detection on credit card transactions using machine learning algorithms. *Iconic Res. And Eng. Journals*, 7(8):293–299, 2024.

[22] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. Data mining for credit card fraud: A comparative study. *Decis. support systems*, 50(3):602–613, 2011.

[23] Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, and Changjun Jiang. Random forest for credit card fraud detection. In *2018 IEEE 15th international conference on networking, sensing and control (ICNSC)*, pages 1–6. IEEE, 2018.

[24] Toluwase Ayobami Olowookere and Olumide Sunday Adewale. A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach. *Sci. Afr.*, 8:e00464, 2020.

[25] Vahid Sinap. Comparative analysis of machine learning techniques for credit card fraud detection: Dealing with imbalanced datasets. *Turkish journal engineering*, 8(2):196–208, 2024.

[26] Nor Aishah Ahad, Friday Zinzendoff Okwonu, Yik Siong Pang, and Olimjon Shukurovich Sharipov. A comparative performance analysis of several machine learning classifiers on the credit card data. *J. Adv. Res. Comput. Appl.*, 37(1):50–64, 2024.

About Authors

Kelechi K. AMAMBA is an accomplished data scientist and machine learning expert with a strong academic foundation and diverse professional experience. He holds advanced degrees in Electrical and Electronics Engineering (BSc, University of Ibadan), Business Administration (MBA, Obafemi Awolowo University), and Business Analytics (MSc, Kent State University). Kelechi has further enhanced his expertise through distinguished certifications in data science and machine learning, including credentials from the MIT Schwarzman College of Computing. His research interests center on artificial intelligence, business and supply chain optimization, energy efficiency, and sustainability. Kelechi brings practical experience in developing predictive analytics solutions, advanced model development, and natural language processing (NLP). He effectively applies data science techniques to solve complex challenges across the biotech, pharmaceutical, financial services, and manufacturing industries, driving innovation and operational excellence.

Olufemi S. OLONILUYI is a Business Analyst, IT Analyst, and AML Financial Analyst with expertise in the IT and financial services sectors. He analyzes business needs, optimizes processes, and develops and implements technology solutions. With a strong background in data analysis, systems integration, and project management, he excels at bridging the gap between business and IT. He is passionate about driving efficiency, fostering innovation, and delivering impactful business solutions. Olufemi holds an Master of Science in Business Analytics from Kent State University, where he gained a solid foundation in business systems, data analytics, and information technology. His educational background complements his professional experience, equipping him with both technical expertise and strategic business insight.

Olayinka H. SIKIRU is a seasoned Payment Operations Senior Specialist with a strong background in data analysis, process optimization, and risk assessment. With extensive experience in financial operations, stakeholder management, and customer-centric service, he has successfully led product improvements and enhanced operational efficiency. Olayinka holds a Master's in Business Analytics from Kent State University and a Bachelor's in Business and Computer Science. His expertise spans SQL, R, Python, and strategic planning, with multiple professional certifications. He is Passionate about problem-solving and leadership. And he thrives in high-impact roles that drive innovation and excellence in financial services.