SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X VOLUME 4, ISSUE 6, 2026

BAYESIAN NETWORK AND TREE-BASED CLASSIFIERS FOR CREDIT CARD FRAUD DETECTION: A COMPARATIVE STUDY ON RAW AND TRANSFORMED TRANSACTION DATA

Waleed Khan[*1], Muqqadus Bibi[2], Sarang Ahmed[3], Muhammad Tahir[4]

[1]*Department of Computer Science, SZABIST University, Karachi, Sindh, Pakistan*
[2]*Department of Robotics & Artificial Intelligence, SZABIST University, Karachi, Sindh, Pakistan*
[3,4]*Department of Computer Science, Faculty of Engineering Science and Technology (FEST), Iqra University Main Campus, Defence View, Karachi – City 75500, Sindh, Pakistan*

[*3]muhammad.tahir01@iqra.edu.pk

DOI:https://doi.org/10.5281/zenodo.21028178

Keywords
Credit Card Fraud Detection; Bayesian Network; Naive Bayes; TAN; Logistic Regression; J48; WEKA; PCA; Feature Engineering; Data Preprocessing

Article History
Received: 25 April 2026
Accepted: 04 June 2026
Published: 21 June 2026

Corresponding Author: *
Waleed Khan

Abstract
Credit card fraud detection remains a critical concern for financial institutions, which face billions of dollars in annual losses from fraudulent transactions. Traditional rule-based detection methods struggle to keep pace with the constantly evolving tactics used by fraudsters, motivating the adoption of Machine Learning (ML) approaches that can automatically learn discriminative patterns from transaction data. This study evaluates five classification algorithms first, K2, Naïve Bayesian, second, Tree-Augmented Naive Bayes (TAN), third, Logistic Regression, and fifth, J48 decision tree – for detecting fraudulent credit card transactions using the WEKA data mining tool with 10-fold cross-validation. Two experiments were conducted: the first applied the classifiers directly to a raw dummy transaction dataset, while the second applied the same classifiers after data transformation and Principal Component Analysis (PCA)-based dimensionality reduction. Results show a substantial performance gain after preprocessing: classifier accuracy rose from a range of 41.8%–84.0% on the raw dataset to 95.8%–100% on the transformed dataset, while false positive rates fell sharply across all models. Logistic Regression and J48 achieved the strongest overall performance on the transformed dataset, each reaching 100% accuracy, precision, recall, and F-measure. These findings confirm that rigorous data preprocessing and dimensionality reduction are decisive factors in building reliable, low-false-alarm credit card fraud detection systems.

I. INTRODUCTION

Credit card fraud is a major concern for financial institutions and their consumers worldwide. Fraud occurs when criminals use stolen or falsified credit card data to conduct illegal transactions, causing financial losses to banks and other financial institutions that are estimated to run into the billions of dollars annually. Consumers also face disruption and potential financial burden from fraudulent activity on their accounts. By 2020, credit card fraud accounted for 30.4% of all identity theft cases in the United States, making it the most common form of fraud reported to the Federal Trade Commission (FTC, 2021).

Financial institutions are increasingly allocating resources toward robust fraud detection systems to reduce these losses and protect consumers. Traditional fraud detection methods rely on pre-established rules and professional expertise to flag suspicious transactions; however, these methods

often prove inadequate because fraud patterns continuously evolve. Fraudsters constantly adapt their tactics to circumvent existing detection systems, which makes sophisticated, adaptive techniques necessary to stay ahead of emerging threats.

A. Problem Statement

The limitations of traditional, rule-based fraud detection methods emphasize the importance of upgrading these systems. Machine learning (ML) has become a powerful tool in this context, allowing automatic analysis of patterns across large datasets to identify anomalies that may indicate fraudulent activity. A key challenge is class imbalance: legitimate transactions vastly outnumber fraudulent ones, which biases models toward the majority class and makes it harder to detect genuine fraud. Building machine learning models that perform well at fraud detection requires advanced algorithms capable of identifying and constructing informative features from transactional data. Effective feature engineering, including the derivation of new or modified inputs, can reveal subtle patterns and relationships that indicate fraudulent activity and thereby improve the overall effectiveness of the detection system.

B. Objectives

The main objective of this research is to evaluate how Bayesian network-based and tree-based classifiers perform on credit card transaction data, and to determine how data preprocessing and dimensionality reduction affect their ability to distinguish fraudulent from legitimate transactions. Specific objectives are:

1) Identify and apply classification algorithms (K2, Naïve Bayesian, TAN, Logistic Regression, and J48) suited to credit card fraud detection.

2) Develop a structured preprocessing pipeline, including data transformation and Principal Component Analysis (PCA), to improve model discriminative power.

3) Evaluate and compare classifier performance on raw versus preprocessed (transformed) transaction data using standard classification metrics.

4) Determine which classifiers offer the best trade-off between detection accuracy, false alarm rate, and computational efficiency for practical deployment.

C. Research Questions

1) Which classification algorithms achieve the strongest performance for credit card fraud detection on raw transaction data?

2) How does data preprocessing and dimensionality reduction (PCA) affect classifier accuracy, precision, recall, and false positive rate?

3) What is the trade-off between detection accuracy and processing speed across the evaluated classifiers?

4) How do Bayesian network-based classifiers compare to tree-based and regression-based classifiers after preprocessing?

D. Significance of the Study

Improved fraud detection has a significant and far-reaching impact for financial institutions and consumers alike. A more accurate detection system can produce substantial cost savings by reducing losses from fraudulent activity, while also strengthening consumer trust through better protection against unauthorized transactions. The methodology and findings of this study extend beyond the immediate context of credit card fraud: the same combination of feature engineering, dimensionality reduction, and classifier comparison is broadly applicable to related anomaly-detection domains such as cybersecurity, healthcare claims auditing, and insurance fraud detection. By rigorously comparing classifier performance before and after preprocessing, this study contributes practical guidance for financial institutions and researchers seeking reliable, low-false-alarm fraud detection pipelines.

II. LITERATURE REVIEW

The threat of credit card fraud in the evolving landscape of financial transactions has led to extensive exploration of detection techniques that harness data mining and machine learning. Several studies have focused on improving detection and reducing false alarms. One notable research effort using self-organizing map neural networks obtained a receiver operating characteristic (ROC) score

above 95%, indicating a high detection rate with comparatively few false alarms. Among machine learning models, the Hidden Markov Model (HMM) has also been used for credit card fraud detection, producing low false alarm rates, though its effectiveness varies across different transaction environments.

Stolfo et al. [10] examined a meta-learning approach using four base classifiers — Iterative Dichotomiser 3 (ID3), Classification and Regression Tree (CART), Ripper, and a Bayesian classifier — to build a robust fraud detection framework. Their study, which balanced the dataset between fraudulent and non-fraudulent cases, found that a Bayesian meta-learner outperformed other meta-learning configurations. However, the artificial 50/50 balance between fraud and non-fraud cases raises questions about real-world applicability, since legitimate transactions vastly outnumber fraudulent ones in practice, suggesting that reported performance may be optimistic relative to deployment conditions.

Bayesian networks in particular have received sustained attention in the literature on credit card fraud detection due to their ability to model complex probabilistic dependencies. Maes et al. [12] compared Bayesian Belief Networks (BBN) and Artificial Neural Networks (ANN) for credit card fraud detection and found that the Bayesian network outperformed the neural network by approximately 8%, while also requiring less classification processing time. Separately, Bahnsen et al. [13] proposed a Bayes minimum risk approach that incorporated cost parameters directly into model optimization; when applied to a Random Forest classifier, this approach produced improved results compared to Logistic Regression (LR) and C4.5 models.

Dal Pozzolo et al. (2018) emphasized the importance of addressing data imbalance and proposed a cost-sensitive learning strategy that led to measurable improvements in detection performance. While these studies recognize the importance of feature engineering and cost-sensitive learning, the present study extends this line of work by systematically comparing Bayesian network-based classifiers (K2, Naïve Bayesian, TAN) against tree-based (J48) and regression-based (Logistic Regression) classifiers under both raw and preprocessed data conditions, providing a direct, controlled comparison of how preprocessing affects each model family.

A. Bayesian Network Classifiers

A Bayesian Network (BN) is a directed acyclic graph in which each node represents a random variable in the dataset and each arc represents a probabilistic dependency between two variables. Bayesian networks can compute the conditional probability of a node given the values of its parent nodes, and offer the advantage of handling asymmetric inputs while making causal relationships explicit [17].

Naïve Bayes (NB) is the simplest and most widely used Bayesian classifier. It is a probabilistic classifier based on Bayes' theorem that makes a strong independence assumption: the presence or absence of a given feature is assumed to be unrelated to the presence or absence of any other feature, given the class. This independence assumption makes Naïve Bayes efficient and easy to implement, though it can be unrealistic for transaction data in which features are often correlated.

Tree-Augmented Naive Bayes (TAN) relaxes the strict independence assumption of Naïve Bayes by allowing limited dependencies between features, represented as a tree structure layered on top of the naive Bayes graph. TAN uses a Bayesian scoring function to determine which feature dependencies to model, allowing it to capture some of the correlation between features that Naïve Bayes ignores.

K2 is a Bayesian network structure-learning algorithm that uses a greedy, hill-climbing search guided by a scoring function to determine the most probable network structure given the data and a specified variable ordering [18]. Because K2 relies on a fixed initial ordering and a greedy search, its performance can be sensitive to how well that ordering reflects the true causal structure of the data.

B. Logistic Regression and J48 Decision Tree

Logistic Regression is a discriminative classifier that directly models the conditional probability of the class label given the input features, using a logistic (sigmoid) link function. Unlike Naïve Bayes, which

is a generative model based on independence assumptions, Logistic Regression estimates class probabilities through iterative likelihood maximization, which often makes it more robust when features are correlated.

J48, available in the WEKA toolkit, is an open-source Java implementation of the C4.5 decision tree algorithm developed by Quinlan [19]. C4.5 builds a decision tree from labeled data by recursively splitting on the feature that yields the greatest information gain, producing an interpretable model in which internal nodes represent feature tests, branches represent feature values, and leaf nodes represent the predicted class. Each of these five classifiers — K2, Naïve Bayesian, TAN, Logistic Regression, and J48 — embodies a different modeling assumption, ranging from strict feature independence (Naïve Bayes) to flexible dependency structures (TAN, K2) to discriminative probability estimation (Logistic Regression) and rule-based partitioning (J48). Comparing all five under identical experimental conditions, both before and after preprocessing, provides insight into which modeling assumptions are best suited to credit card transaction data.

III. METHODOLOGY

A. Data Collection

The dataset used for this analysis is "creditcard.csv," which contains credit card transactions made by European cardholders in September 2013. The dataset contains 284,807 transactions recorded over two days, of which 492 transactions (about 0.172% of the total) are fraudulent. Most input variables are anonymized principal components labeled V1 through V28, produced by Principal Component Analysis (PCA) for confidentiality reasons. Two additional attributes are not subject to PCA transformation: Time (the number of seconds elapsed between a given transaction and the first transaction in the dataset) and Amount (the monetary value of the transaction). The target Class label indicates whether a transaction is fraudulent (1) or legitimate (0).

B. Data Preprocessing

Data preprocessing is an essential step for ensuring the accuracy and reliability of the analysis. The dataset was confirmed to be free of missing values by examining the count of non-null entries in each column. The Time and Amount features were standardized using the StandardScaler function from the scikit-learn library so that they would be on the same scale as the PCA-transformed features, with a resulting mean of 0 and standard deviation of 1. The dataset was then partitioned into training and testing sets using an 80:20 split, with stratification used to preserve the original class distribution in both partitions.

C. Feature Engineering

Because most features were already PCA-transformed, feature engineering focused on enriching the Time and Amount attributes and deriving new behavioral and anomaly-related features:

Time-related features: the hour of day was extracted from the Time feature to capture daily transaction patterns; a binary day/night indicator flagged whether a transaction occurred between 6 AM and 6 PM or outside that window; and a time-delta feature captured the interval between consecutive transactions for each cardholder to identify frequency-based patterns.

Behavioral features: transaction frequency was computed within defined time windows (e.g., past hour, past day), and statistical descriptors of transaction amount — mean, standard deviation, and coefficient of variation — were used to flag irregular spending patterns.

Anomaly features: high-value transactions were flagged when they deviated substantially from a cardholder's average transaction amount, and rapid successive transactions were flagged as a possible indicator of fraud.

D. Synthetic Dataset Construction and Labeling Rules

Because access to real, unmasked credit card fraud data is restricted by strict privacy regulations, a synthetic dummy dataset was additionally constructed to validate the modeling pipeline under controlled, interpretable conditions, following the labeling scheme summarized in **Table 1**. Attributes such as PIN similarity to a blacklisted PIN, whether the transaction originated from a

blacklisted country, whether a usage threshold was exceeded, the time interval between attempts, and address similarity were combined to determine whether a case was labeled fraudulent.

Table 1. Rules for Labeling Fraudulent Transactions (T = TRUE, F = FALSE)

Case	Similar PIN	Blacklisted Country	Exceed Threshold	Time Interval	Similar Address	Fraud
1	T	T	T	T	T	T
2	T	T	T	T	F	T
3	T	T	T	F	F	T
4	T	T	F	T	F	T
5	T	T	F	F	T	T
6	T	F	T	T	T	T
7	T	F	T	F	F	T
8	T	F	T	T	F	T
9	T/F	T/F	T/F	T/F	T/F	T

Table 1 shows the rule set used to label synthetic transactions as fraudulent. The dummy data were generated using a spreadsheet-based generator (generatedata.com) together with a custom GNU scripting pipeline, producing attributes such as reference number, terminal ID, PIN entered, transaction value, timestamp, location, and shipping address.

E. Classification and Evaluation Setup

Classifier performance was evaluated in WEKA, a widely used open-source data mining and machine learning workbench applied to tasks such as sentiment analysis, identity recognition, spam filtering, and fraud detection. All classifiers were evaluated using 10-fold cross-validation, in which the dataset is partitioned into ten folds; in each iteration nine folds are used for training and the remaining fold for testing, and the results are averaged across all ten iterations. This procedure ensures that every observation is used for both training and testing exactly once, improving the reliability of the reported metrics.

Performance was assessed using standard classification metrics derived from the confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) counts. From these, True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, F-Measure, and Accuracy were computed, as defined in **Table 2**.

Table 2. Definitions and Formulas of Evaluation Metrics

Metric	Formula
True Positive Rate (TPR)	TPR = TP / (TP + FN)
False Positive Rate (FPR)	FPR = FP / (FP + TN)
Precision	Precision = TP / (TP + FP)
Recall	Recall = TP / (TP + FN)
F-Measure	F-Measure = 2×TP / (2×TP + FP + FN)

Metric	Formula
Accuracy	Accuracy = (TP + TN) / (TP + TN + FP + FN)

Two experiments were conducted using this evaluation setup. Experiment 1 applied all five classifiers directly to the raw dummy dataset. Experiment 2 applied the same classifiers after data transformation and PCA-based dimensionality reduction, which also reduced the influence of less informative attributes such as terminal_id. **Figure 1** summarizes the overall workflow connecting data preprocessing, model training, and verification.

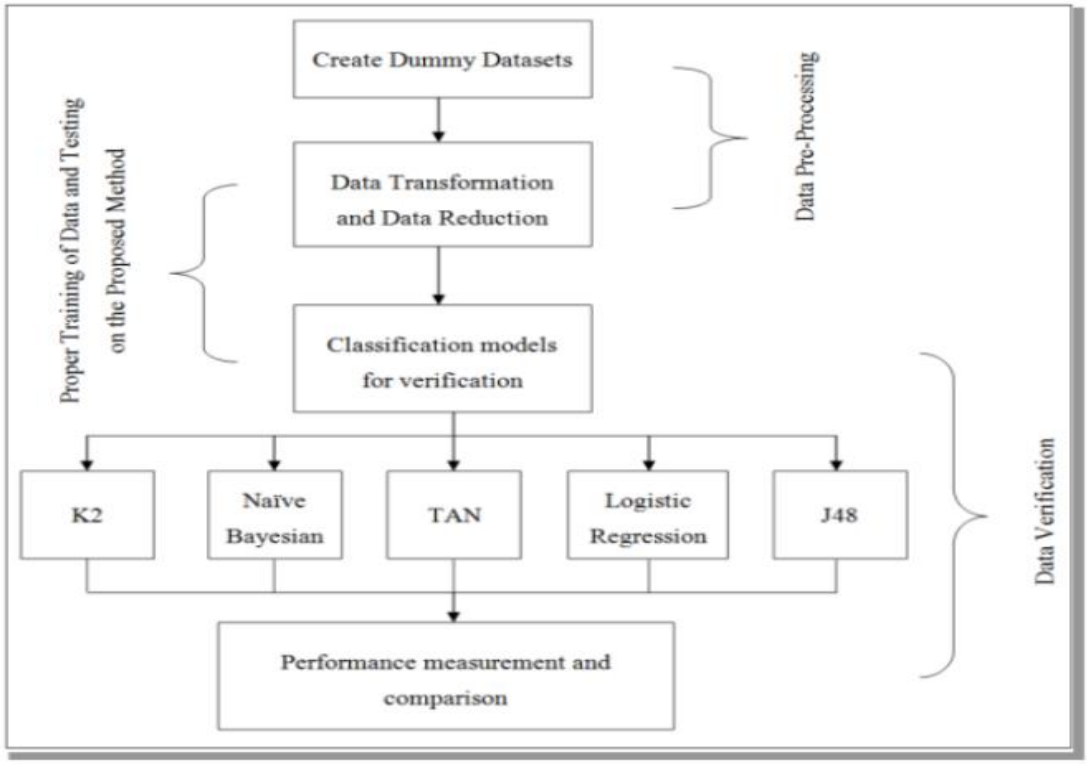

Figure 1. Workflow for credit card fraud detection model development and verification, from dummy dataset creation through data preprocessing to classifier training and performance comparison.

Figure 1 illustrates the end-to-end methodology. Dummy datasets are first created and then subjected to data transformation and reduction. The five classifiers — K2, Naïve Bayesian, TAN, Logistic Regression, and J48 — are trained on the resulting data, and their outputs are passed to a final performance measurement and comparison stage. This staged design allows the effect of preprocessing to be isolated by comparing classifier performance before and after the transformation step.

IV. RESULTS AND DISCUSSION

A. Experiment 1: Classification on the Raw Dataset

Table 3. Classification Results Using the Raw Dummy Dataset

Metric	K2	Naïve Bayesian	TAN	Logistic Regression	J48
True Positive Rate (%)	31.0	50.3	75.0	60.3	73.0

Metric	K2	Naïve Bayesian	TAN	Logistic Regression	J48
False Positive Rate (%)	69.0	49.7	25.0	39.7	27.0
Precision (%)	21.0	45.7	73.0	44.7	69.4
Recall (%)	32.0	60.3	75.0	47.8	67.5
F-Measure (%)	32.2	34.3	68.5	44.9	67.4
Processing Speed (s)	10.0	10.0	56.0	25.0	84.0
Accuracy (%)	41.8	53.7	84.0	67.3	80.0

Table 3 reports the results of Experiment 1, in which all five classifiers were trained and evaluated directly on the raw dummy dataset without transformation. TAN achieved the strongest overall performance, with the highest TPR (75.0%), precision (73.0%), recall (75.0%), F-measure (68.5%), and accuracy (84.0%), while also maintaining a comparatively low false positive rate. J48 followed closely, with slightly lower metrics but a markedly slower processing speed (84.0 seconds) due to the cost of building and pruning its tree structure. Logistic Regression, Naïve Bayesian, and especially K2 underperformed relative to TAN and J48, with K2 recording the lowest accuracy (41.8%) and the highest false positive rate (69.0%). This pattern suggests that noise and unrefined attributes in the raw data disproportionately affect classifiers that rely on simplifying independence assumptions or fixed variable orderings.

B. Experiment 2: Classification on the Transformed Dataset

Table 4. Classification Results Using the Transformed (PCA-Reduced) Dataset

Metric	K2	Naïve Bayesian	TAN	Logistic Regression	J48
True Positive Rate (%)	91.7	99.6	99.7	100.0	100.0
False Positive Rate (%)	8.3	0.4	0.3	0.0	0.0
Precision (%)	92.6	95.6	98.4	100.0	100.0
Recall (%)	91.7	99.6	99.6	100.0	100.0
F-Measure (%)	95.7	89.3	99.0	100.0	100.0
Processing Speed (s)	2.0	2.0	30.0	5.0	32.0
Accuracy (%)	95.8	96.7	99.7	100.0	100.0

Table 4 reports the results of Experiment 2, in which the same five classifiers were evaluated after data transformation and PCA-based dimensionality reduction. Every classifier improved substantially relative to Experiment 1. Logistic Regression and J48 reached 100% across TPR, precision, recall, F-measure, and accuracy, while TAN closely followed with 99.7% accuracy and a 0.3% false positive rate. Even K2, the weakest performer in Experiment 1, improved to 95.8% accuracy with an 8.3% false positive rate. Processing speed also improved markedly for every classifier — for example, J48 dropped from 84.0 seconds to 32.0 seconds — indicating that PCA-based dimensionality reduction not only improved discriminative power but also reduced the computational burden of training.

C. Visual Comparison of Classifier Performance

To make the effect of preprocessing easier to interpret, the accuracy, error-rate, and processing-

speed results from **Tables 3** and 4 are visualized in **Figures 2** through 4.

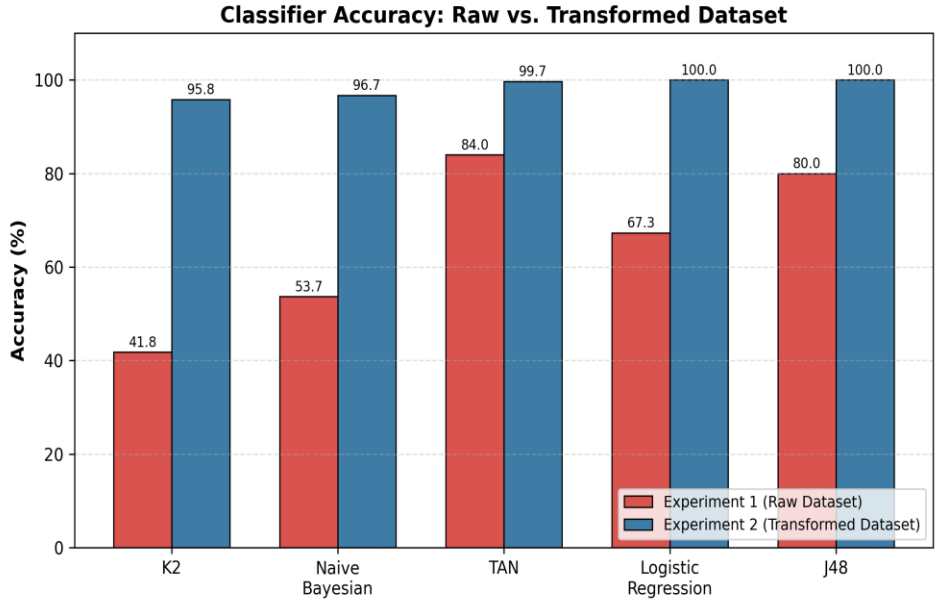

Figure 2. Classifier accuracy on the raw dataset (Experiment 1) versus the transformed dataset (Experiment 2).

Figure 2 shows the accuracy gain achieved by every classifier after preprocessing. The gain is largest for K2 (41.8% → 95.8%, a 54-point increase) and Naïve Bayesian (53.7% → 96.7%), confirming that classifiers relying on simplifying probabilistic assumptions benefit most from a cleaner, lower-dimensional feature space. TAN, Logistic Regression, and J48 already performed reasonably well on raw data and reached near-perfect or perfect accuracy after transformation, indicating that PCA-based preprocessing benefits all model families but is especially decisive for the weaker baseline classifiers.

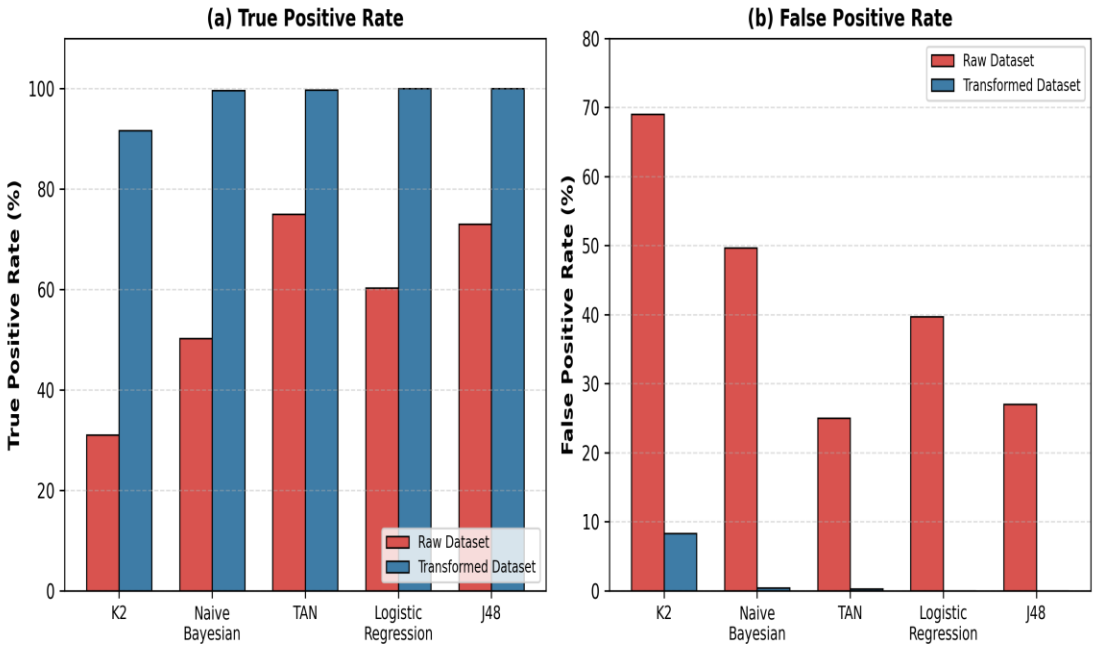

Figure 3. True positive rate and false positive rate for each classifier, comparing raw and transformed datasets.

Figure 3 separates detection sensitivity from false-alarm behavior. Panel (a) shows that true positive rates rose substantially for every classifier after transformation, most notably for K2 (31.0% → 91.7%). Panel (b) shows an even sharper reduction in false positive rate: K2's FPR fell from 69.0% to 8.3%, while Logistic Regression and J48 reduced their false positive rate to effectively 0%. Because a high false positive rate in production would mean flagging large volumes of legitimate transactions as fraudulent, this reduction is arguably the most operationally significant outcome of the preprocessing pipeline, as it directly determines how much manual review burden a deployed system would place on a financial institution's fraud team.

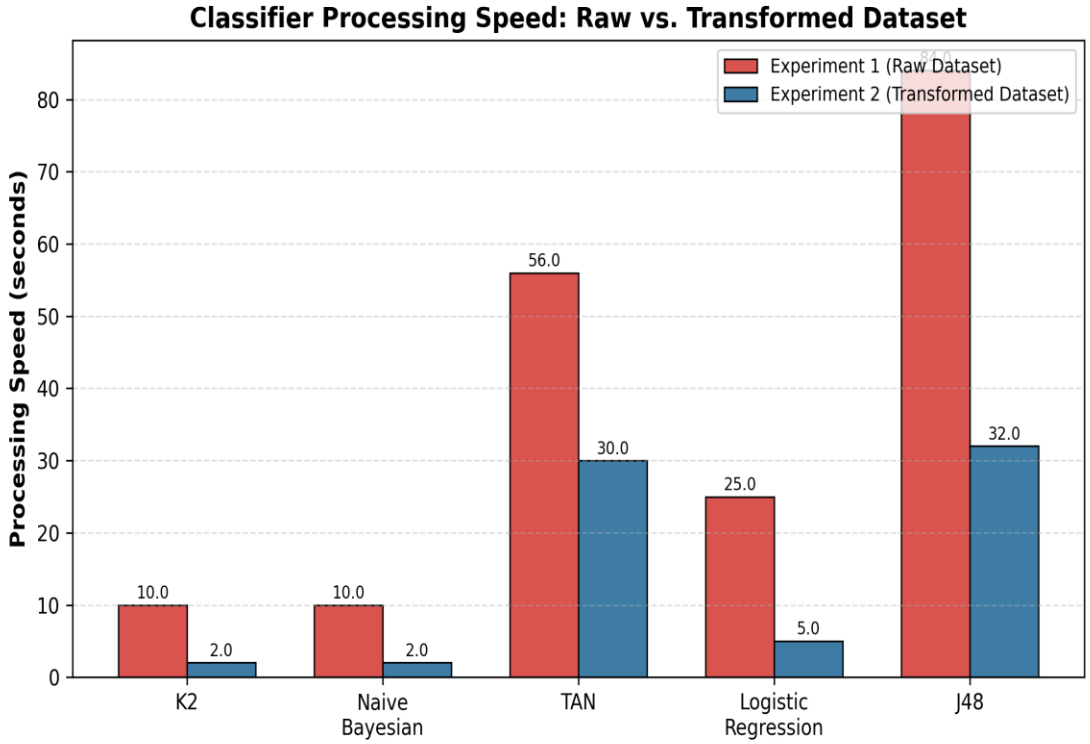

Figure 4. Classifier processing speed (seconds) on the raw dataset versus the transformed dataset.

Figure 4 shows that dimensionality reduction also reduced training and evaluation time for every classifier. K2 and Naïve Bayesian processed the transformed data in 2.0 seconds (down from 10.0 seconds), TAN dropped from 56.0 to 30.0 seconds, Logistic Regression from 25.0 to 5.0 seconds, and J48 — the slowest classifier overall — from 84.0 to 32.0 seconds. This consistent speed-up reinforces that PCA-based preprocessing is not a pure accuracy-versus-cost trade-off in this study: it simultaneously improved detection performance and reduced computational overhead, which is a favorable combination for real-time fraud detection systems where both accuracy and latency matter.

D. Discussion

Taken together, the results in Tables 3 and 4 and Figures 2 through 4 show that data preprocessing and PCA-based dimensionality reduction are decisive factors in credit card fraud detection performance, often mattering more than the choice of classifier itself. On raw data, model family clearly affected performance, with TAN and J48 outperforming K2, Naïve Bayesian, and Logistic Regression. After preprocessing, these differences largely disappeared, with Logistic Regression and J48 reaching perfect scores and the remaining classifiers close behind. This suggests that much of the raw-data performance gap was attributable to noisy or redundant attributes (such as terminal_id) rather than to fundamental limitations of the weaker classifiers' modeling assumptions.

These findings are consistent with prior work emphasizing the importance of feature engineering and cost-sensitive learning in fraud detection [12], [13]. However, the magnitude of improvement observed here — particularly the near-total elimination of false positives for Logistic Regression and J48 — underscores how much practical fraud detection performance depends on the preprocessing pipeline. A limitation of this study is that the underlying creditcard.csv dataset is itself already PCA-anonymized by its original publishers, and part of the experimentation relies on a synthetic dummy dataset constructed to mimic realistic fraud patterns under controlled, interpretable labeling rules. Real-time, large-scale validation of this preprocessing pipeline on

unmasked, production-scale transaction data remains an important direction for future work.

V. CONCLUSION AND FUTURE WORK

This study compared five classification algorithms — K2, Naïve Bayesian, TAN, Logistic Regression, and J48 — for credit card fraud detection, evaluated using WEKA with 10-fold cross-validation under two conditions: a raw dummy transaction dataset and the same dataset after transformation and PCA-based dimensionality reduction. On raw data, TAN and J48 were the strongest performers, while K2 lagged substantially behind. After preprocessing, all classifiers improved markedly, with Logistic Regression and J48 reaching 100% accuracy, precision, recall, and F-measure, and even the weakest raw-data performer, K2, reaching 95.8% accuracy. Preprocessing also reduced processing time for every classifier, demonstrating that a well-designed preprocessing pipeline can simultaneously improve detection accuracy, reduce false alarms, and lower computational cost.

Future work should extend this evaluation to real-time, production-scale transaction streams rather than dummy or fully anonymized data, in order to validate that the observed gains transfer to live deployment conditions. Given the strong performance of Bayesian network-based and tree-based classifiers observed here, future studies could also contrast these approaches with hyperplane-based classifiers such as Support Vector Machines, and explore hybrid or ensemble architectures that combine the interpretability of Bayesian and tree-based models with the robustness of discriminative classifiers.

REFERENCES

WorldPay. (2015, Nov.). Global payments report preview: Your definitive guide to the world of online payments. Retrieved September 28, 2016.

Federal Trade Commission. (2008). Consumer Sentinel Network data book for January–December 2008. Retrieved October 20, 2016, from https://www.ftc.gov/.

T. P. Bhatla, V. Prabhu, and A. Dua, "Understanding credit card frauds," Cards Business Review #2003-1, Tata Consultancy Services, 2003.

The Nilson Report. (2015). Global fraud losses reach $16.31 billion. Issue 1068, July 2015.

Y. Sahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in Proc. Int. MultiConf. Engineers and Computer Scientists (IMECS), vol. I, March 2011.

C. Elkan, "Magical thinking in data mining: Lessons from COIL Challenge 2000," in Proc. SIGKDD'01, 2001, pp. 426–431.

M. J. Zaki and W. Meira Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge, UK: Cambridge University Press, 2014.

F. N. Ogwueleka, "Data mining application in credit card fraud detection system," J. Eng. Sci. Technol., vol. 6, no. 3, pp. 311–322, 2011.

V. Bhusari and S. Patil, "Application of hidden Markov model in credit card fraud detection," Int. J. Distributed Parallel Syst., vol. 2, no. 6, 2011.

S. J. Stolfo, D. W. Fan, W. Lee, A. L. Prodromidis, and P. K. Chan, "Credit card fraud detection using meta-learning: Issues and initial results," in Proc. AAAI Workshop AI Methods in Fraud and Risk Management, 1998, pp. 83–90.

S. K. Sen and S. Dash, "Meta-learning algorithms for credit card fraud detection," Int. J. Eng. Res. Develop., vol. 6, no. 6, pp. 16–20, 2013.

S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using Bayesian and neural networks," in Proc. 1st NAISO Congr. Neuro Fuzzy Technologies, Havana, 2002.

A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost sensitive credit card fraud detection using Bayes minimum risk," in Proc. 12th Int. Conf. Machine Learning and Applications, 2013.

A. Kundu, S. Panigrahi, S. Sural, and A. K. Majumdar, "Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning," Inf. Fusion Comput. Security, vol. 10, no. 4, pp. 354–363, 2009.

W. Lam and F. Bacchus, "Learning Bayesian belief networks: An approach based on the MDL principle," Computational Intelligence, vol. 10, no. 3, pp. 269–293, 1994.

M. Mehdi, S. Zair, A. Anou, and M. Bensebti, "A Bayesian networks in intrusion detection systems," Int. J. Comput. Intell. Res., vol. 3, no. 1, 2007.

R. Najafi and M. Afsharchi, "Network intrusion detection using tree augmented naive-Bayes," in Proc. 3rd Int. Conf. Contemporary Issues in Computer and Information Sciences (CICIS), 2012.

G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," Machine Learning, vol. 9, no. 4, pp. 309–347, 1992.

J. R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.

N. Friedman and M. Goldszmidt, "Building classifiers using Bayesian networks," in Proc. 13th National Conf. Artificial Intelligence, vol. 2, 1996, pp. 1277–1284.

N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," Machine Learning, vol. 29, pp. 131–163, 1997.