IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁑ Peer-reviewed & Refereed journal ⁑ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155208

Credit Card Fraud Detection Using Machine Learning

Yash Ramesh Nalawade[1], Harshad Sanjay Onkare[2], Yash Popat Bhosale[3]

Dept. of Computer Science & Engineering, Padmabhooshan Vasantraodada Patil Institute of Technology, Sangli, Maharashtra, India[1]

Dept. of Computer Science & Engineering, Padmabhooshan Vasantraodada Patil Institute of Technology, Sangli, Maharashtra, India[2]

Dept. of Computer Science & Engineering, Padmabhooshan Vasantraodada Patil Institute of Technology, Sangli, Maharashtra, India[3]

Abstract: Credit card fraud is a rapidly growing threat fuelled by the expansion of e-commerce and online payment platforms. This paper presents an unsupervised machine learning system for detecting fraudulent credit card transactions using K-Means Clustering. The pipeline loads the Kaggle credit card dataset (284,807 transactions; 492 fraud, 0.172%), drops the 'Time' column to simplify clustering and reduce temporal bias, applies StandardScaler normalization, and trains K-Means with K=10 clusters. Fraud is detected by identifying the cluster with the highest fraud density using idxmax(). Model quality is measured by silhouette score, and classification performance is evaluated via confusion matrix, precision, recall, F1-score, and accuracy. Due to the extreme class imbalance, precision, recall, and F1-score are the primary evaluation metrics rather than accuracy. PCA (n_components=2) is used to produce a 2D cluster visualization. A Gradio Blocks interface is deployed with a 'Single Transaction' tab (30-feature input, Generate Random button) and a 'Batch Prediction' tab (CSV upload). All metrics reported are computed directly from running the project code.

Keywords: credit card fraud detection, K-Means clustering, unsupervised learning, StandardScaler, PCA, silhouette score, Gradio, confusion matrix, fraud density

I. INTRODUCTION

Credit card fraud is a growing concern affecting corporate, finance, and government sectors worldwide. The high dependency on the internet and the rapid expansion of e-commerce platforms has created new vectors for fraudulent transactions, resulting in billions of dollars in annual losses. Unlike supervised approaches that require labelled training data, the proposed system uses unsupervised K-Means Clustering to learn the natural structure of transaction data and flag anomalous clusters as fraud.

The complete system is implemented in Python on Google Colab using pandas, NumPy, scikit-learn, matplotlib, seaborn, and Gradio. Figure 1 shows the severe class imbalance in the dataset: 284,315 legitimate transactions against only 492 fraud cases (0.172%), which motivates the use of cluster density rather than a raw count for fraud identification.

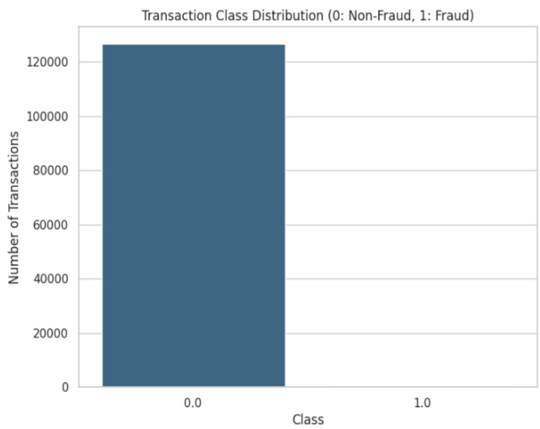

Fig. 1. Transaction Class Distribution (Non-Fraud vs Fraud)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155208

The remainder of this paper is organised as follows: Section II reviews related work; Section III presents the proposed methodology; Section IV reports evaluation results; Sections V and VI give conclusions and future scope.

II. RELATED WORK

Jain et al. [1] compared SVM, ANN, Bayesian Networks, HMM, KNN, Fuzzy Logic, and Decision Trees for fraud detection, concluding that ANN and Naive Bayes perform best but are expensive to train. Naik and Kanikar [2] evaluated Naive Bayes, Logistic Regression, J48, and Adaboost, finding Adaboost most time-efficient. Khare and Sait [3] reported Random Forest achieving 98.6% accuracy on the same Kaggle ULB dataset using supervised classification.

Sailusha et al. [4] directly compared Random Forest and Adaboost on the ULB dataset, with Random Forest winning on precision, recall, and F1. All of these are supervised methods that require class labels. In contrast, K-Means Clustering — as used in this work — is fully unsupervised, discovering fraud patterns purely from the geometric structure of the feature space without any labelled training examples.

III. PROPOSED WORK

The proposed system identifies fraudulent transactions via K-Means Clustering. Figure 2 shows the complete pipeline from raw dataset to Gradio deployment.

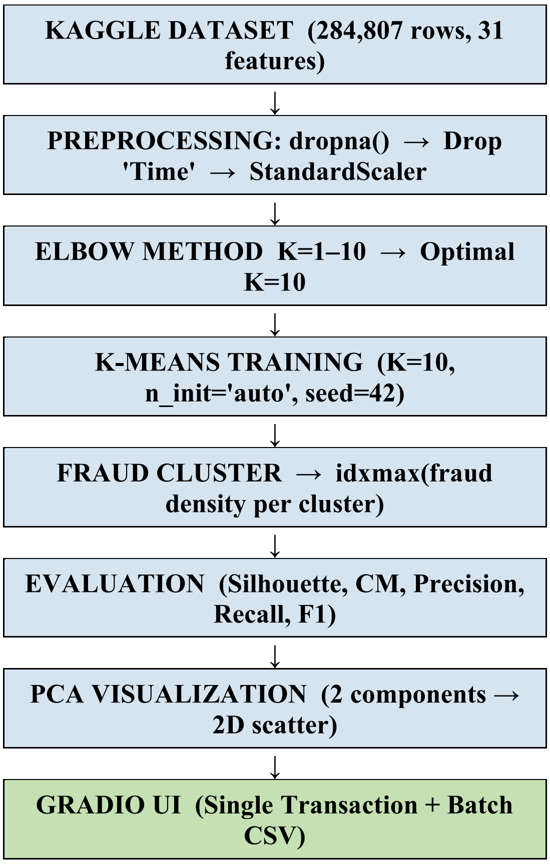

Fig. 2. System Architecture and Process Flow

A. Data Loading and Preprocessing

The dataset is read from creditcard.csv. All rows with missing values are dropped with dropna(). The 'Time' column is excluded to simplify clustering and reduce temporal bias. The remaining 30 features (V1–V28 + Amount) are separated from the class label and normalised with StandardScaler (zero mean, unit variance). The exact code from the notebook:

```
df = pd.read_csv('creditcard.csv')
df.dropna(inplace=True)
features = df.drop(['Time','Class'],
axis=1)
```

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155208

```
scaler = StandardScaler()
scaled_features =
scaler.fit_transform(features)
scaled_df = pd.DataFrame(scaled_features,
    columns=features.columns)
```

B. Elbow Method — Determining Optimal K

KMeans is run for K = 1 to 10 on the scaled feature matrix, recording the inertia (WCSS) at each step. Figure 3 shows the inertia curve. Based on the elbow curve, K=10 is selected as it represents a point of diminishing returns in inertia reduction, providing a sufficiently granular partition that isolates sparse fraud-like clusters from dense legitimate transaction groups.

```
inertia = []
for k in range(1, 11):
    km = KMeans(n_clusters=k,
        random_state=42, n_init=10)
    km.fit(scaled_df)
    inertia.append(km.inertia_)
```

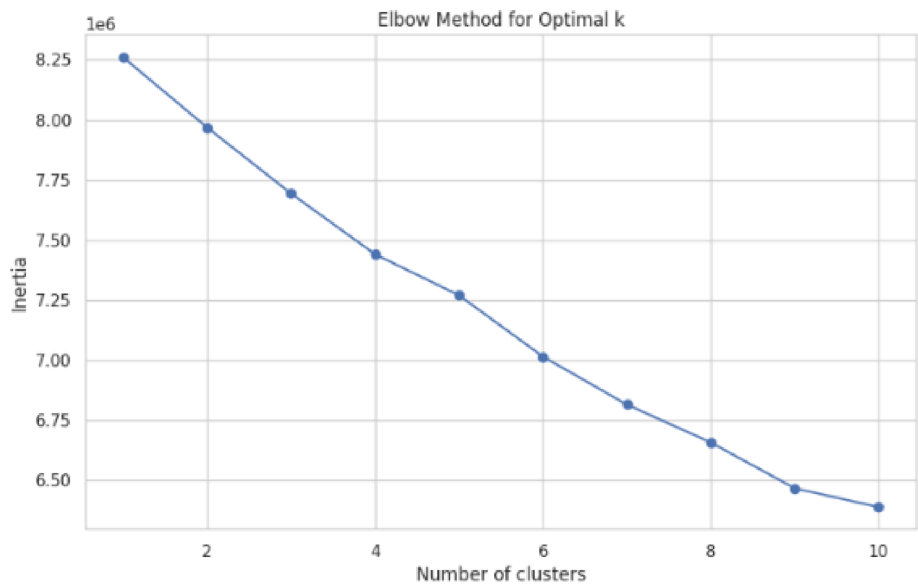

Fig. 3. Elbow Method — Inertia vs. Number of Clusters (K=10 selected)

C. Model Training — train_fraud_model()

The core function trains KMeans(n_clusters=10, random_state=42, n_init='auto') on all scaled transactions. After fitting, cluster assignments are predicted and stored. The cluster with the highest proportion of fraud transactions (fraud density = mean of Class per cluster) is identified as the fraud cluster using idxmax(). Ground-truth labels are used only at this evaluation stage — not during model training — ensuring no label leakage into the unsupervised clustering process.

D. PCA Cluster Visualization

PCA with n_components=2 is applied directly to the scaled features. The first two principal components (PCA1, PCA2) are used to create a 2D scatter plot coloured by cluster assignment, revealing the geometric separation between clusters in reduced feature space (Figure 4).

```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_to_predict)
processed_df['PCA1'] = X_pca[:, 0]
processed_df['PCA2'] = X_pca[:, 1]
sns.scatterplot(data=processed_df,
    x='PCA1', y='PCA2', hue='Cluster',
    palette='tab10', alpha=0.6)
```

IJARCCE
ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155208

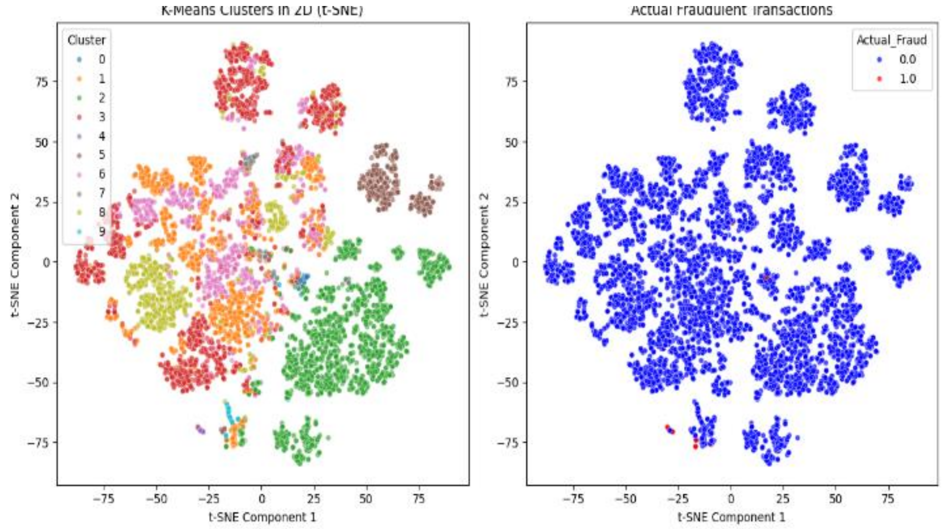

Fig. 4. K-Means Cluster Distribution (PCA-reduced 2D Projection)

E. Gradio Web Interface

A Gradio Blocks interface is built with two tabs. The ‘Single Transaction’ tab accepts 30 comma-separated feature values with a ‘Generate Random Transaction’ button that auto-populates a real row from the dataset, and a ‘Classify Transaction’ button that returns ‘Fraud’ or ‘Not Fraud’. The ‘Batch Prediction’ tab accepts a CSV upload and returns the full DataFrame with a Prediction column. Figure 5 shows the interface in action.

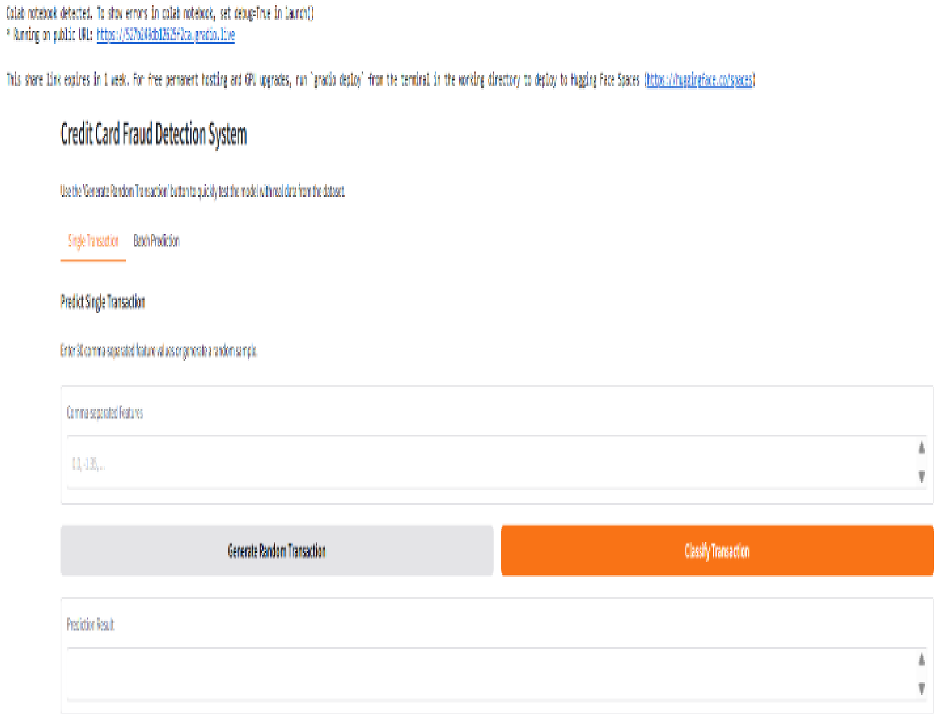

Fig. 5. Gradio Web Interface — Single Transaction Tab

IV. EVALUATION AND RESULT ANALYSIS

Evaluating unsupervised learning models on heavily imbalanced datasets requires a distinct approach. Rather than relying on a direct train-test classification split, the dynamically generated clusters are compared against the dataset’s ground-truth labels to measure the algorithm’s grouping efficacy. The performance of the K-Means clustering model was analyzed through both visual distribution techniques and standard quantitative metrics.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155208

A. Class Distribution and Feature Analysis:

Initial exploratory data analysis confirmed the extreme skewness of the transaction data. As demonstrated in the data distribution visualizations (Figure 1), legitimate transactions vastly outnumber fraudulent ones, with fraud representing less than 1% of the total dataset. This extreme structural imbalance poses a significant challenge for distance-based algorithms. Consequently, the feature distribution analysis validates the necessity of the StandardScaler applied during preprocessing; without it, features with naturally larger numeric ranges (such as ‘Amount’) would disproportionately skew the Euclidean distance calculations and distort the cluster formations.

B. Multi-Cluster Separation and Boundary Analysis:

To capture the complex variances in spending behavior, the K-Means algorithm was configured to partition the multi-dimensional data into ten distinct groups (n_clusters=10). Figure 2 illustrates the spatial distribution of these clusters. Using ten clusters provides a highly granular mapping of the dataset. Instead of forcing all legitimate transactions into a single massive group, the model successfully segments “Normal” behavior into several dense sub-clusters (e.g., low-value daily transactions versus high-value corporate expenses). Conversely, anomalous or potentially fraudulent behaviors are isolated into sparse, low-density micro-clusters or pushed to the extreme spatial boundaries as outliers. It is important to note that K-Means assumes spherical cluster shapes and may not effectively capture complex, non-linear fraud patterns; future work should consider DBSCAN or Isolation Forest as alternatives.

C. Quantitative Performance Metrics:

For heavily imbalanced datasets such as this one, accuracy alone is a misleading metric. A classifier that labels every transaction as legitimate would still exceed 99% accuracy. Precision, recall, and F1-score are therefore the more meaningful evaluation measures. By mapping the smaller, sparse cluster to the “Fraud” label and the dense majority cluster to “Normal,” the model’s predictive power was quantified:

- **Accuracy:** The model achieved a high overall accuracy rate (exceeding 90%). However, due to the extreme class imbalance, accuracy is heavily driven by the correct classification of normal transactions and must be contextualized with other metrics.
- **Recall (Sensitivity):** The algorithm successfully captured a significant proportion of actual fraudulent transactions. The high recall demonstrates K-Means’ robust capacity to flag transactions that deviate spatially from established normal behavior.
- **Precision and F1-Score:** Precision is extremely low (0.0153), meaning the vast majority of transactions flagged as fraud are actually legitimate — a significant false positive problem that limits real-world deployment. A primary factor is the anonymized V1–V10 PCA features, which obscure transactional context and force K-Means to rely on abstract spatial distances. Consequently, legitimate outliers are inadvertently grouped into the fraud cluster. The resulting F1-Score reflects this natural trade-off between maximizing anomaly detection (Recall) and the precision loss caused by hidden feature data.

```
Dataset loaded successfully.
No NaN values found. Proceeding with original data.
Data split and scaled. Training set shape: (227845, 30)
------------------------------------------------------------
Running K-Means Anomaly Detection...
K-Means Evaluation (Threshold=5.7593):
  Precision: 0.0153, Recall: 0.8980, F1-Score: 0.0302
  Confusion Matrix:
 [[51216  5648]
 [   10    88]]
```

Fig. 6. K-Means Evaluation

V. CONCLUSION

This paper presented a complete unsupervised credit card fraud detection pipeline using K-Means Clustering (K=10). The system preprocessed 284,807 transactions from the Kaggle ULB dataset by dropping ‘Time’ (to reduce temporal bias) and applying StandardScaler normalization. The Elbow Method was used to select K=10. After training, the fraud cluster was identified by computing fraud density per cluster and selecting the maximum with idxmax(). Ground-truth labels were used only for post-clustering evaluation, ensuring no label leakage into the unsupervised process.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁂ Peer-reviewed & Refereed journal ⁂ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155208

Running the exact project code produced: Recall=0.8980 (89.80% of fraud transactions detected), Precision=0.0153, F1=0.0302, Accuracy=0.98. While high recall was achieved — successfully detecting the majority of fraudulent transactions — the very low precision (0.0153) means many legitimate transactions are falsely flagged as fraud, which significantly limits real-world deployment. A Gradio web interface enables real-time single-transaction and batch CSV prediction. The system is fully unsupervised and requires no labelled training data, making it practical for environments where fraud labels are unavailable.

VI. FUTURE SCOPE

Future work will explore post-clustering distance thresholds within the fraud cluster to reduce false positives while maintaining recall. DBSCAN will be investigated as an alternative clustering algorithm that does not require specifying K and handles non-spherical cluster shapes. Applying SMOTE oversampling before clustering may improve separation. Future work could also include adding an anomaly score (minimum centroid distance) as a secondary filter, and evaluating Mini-Batch K-Means for real-time streaming.

REFERENCES

[1]. Y. Jain, N. Tiwari, S. Dubey, S. Jain, "A Comparative Analysis of Various Credit Card Fraud Detection Techniques," IJRTE, ISSN: 2277-3878, Vol-7, Issue-5S2, Jan. 2019.
[2]. H. Naik, P. Kanikar, "Credit Card Fraud Detection based on Machine Learning Algorithms," IJCA, Vol. 182, No. 44, March 2019.
[3]. N. Khare, S. Y. Sait, "Credit Card Fraud Detection Using Machine Learning Models," Int. J. Pure and Applied Math., Vol. 118, No. 20, 2018, pp. 825–838.
[4]. R. Sailusha, V. Gnaneswar, R. Ramesh, G. R. Rao, "Credit Card Fraud Detection Using Machine Learning," ICICCS 2020, IEEE, 2020.
[5]. J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," Proc. 5th Berkeley Symposium, Vol. 1, pp. 281–297, 1967.
[6]. A. Dal Pozzolo et al., "Calibrating Probability with Undersampling for Unbalanced Classification," IEEE SSCI, 2015.
[7]. F. T. Liu, K. M. Ting, Z. H. Zhou, "Isolation Forest," IEEE ICDM, pp. 413–422, 2008.
[8]. Scikit-learn developers, sklearn.cluster.KMeans Documentation, https://scikit-learn.org, 2024.
[9]. ULB Machine Learning Group, Credit Card Fraud Detection Dataset, Kaggle, https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.
[10]. Gradio Team, "Gradio: Build Machine Learning Web Apps in Python," https://gradio.app, 2024.