A Machine Learning method with Hybrid Feature Selection for Improved Credit Card Fraud Detection

Problem Statement:

The surge in credit card transactions poses a challenge for fraud detection. This study addresses the issue of redundant and irrelevant features in credit card data, degrading machine learning classifier performance. A hybrid feature-selection technique is proposed to streamline feature relevance, optimizing fraud detection without presenting specific results.

Proposed Work:

A hybrid feature-selection technique is proposed to enhance credit card fraud detection, integrating filter and wrapper methods. Various feature extraction methods and machine learning models, including ELM and ensemble techniques, are explored for improved performance.

System Architecture:

The architecture involves preprocessing credit card transaction data, extracting relevant features, training various ML models, and making predictions on new transactions to detect fraud. Models include ELM, AdaBoost, Logistic Regression, Random Forest, SVM, Decision Tree, and Voting Classifier.

Step By Step Procedure

1. Dataset Collection: Obtain the credit card transaction dataset containing both fraudulent and legitimate transactions. Ensure the dataset is labeled appropriately.

2. Exploratory Data Analysis:

- **Data processing:** Utilizing pandas, the dataset is loaded, missing values are handled, and unnecessary columns are dropped to streamline analysis.
- **Data Normalization:** Min-Max scaling and Standardization techniques are applied to ensure features are on a similar scale, enhancing model stability.
- **Visualization:** Seaborn and Matplotlib are used to generate various plots like histograms, scatter plots, and heatmaps, aiding in understanding data distributions and identifying patterns.

- **Feature Extraction:** Four techniques are employed, including Full Feature, Information Gain (IG), Genetic Algorithm Wrapper (GAW), and a hybrid IG-GAW approach, aiming to select the most relevant features for improved model performance.

3. Dataset Splitting: Split the dataset into training and testing subsets. Typically, 70-80% of the data is used for training and the remaining for testing.

4. Model Development:

- Implement various machine learning models including Extreme Learning Machine (ELM), AdaBoost, Logistic Regression, Random Forest, Support Vector Machine (SVM), Decision Tree, and Voting Classifier.
- Train each model using the training dataset and evaluate their performance using appropriate metrics such as accuracy, precision, recall, and F1-score.

5. Ensemble Approach:

- Implement an ensemble approach to combine predictions from multiple models for a more robust and accurate final prediction.
- Train the ensemble model using predictions from individual models.

6. Testing and Evaluation:

- Use the testing dataset to evaluate the performance of the trained models and the ensemble approach.
- Calculate performance metrics to assess the effectiveness of each model in detecting fraudulent transactions.

7. Visualization and Interpretation: Visualize the results using matplotlib/seaborn to gain insights into model performance and the characteristics of fraudulent transactions.

8. Deployment and Monitoring:

- Deploy the trained models in a production environment for real-time fraud detection.
- Monitor the performance of the deployed models and update them periodically as needed.