Sanjay Saini

Credit Card Fraud Detection: A Comparative Analysis of Traditional Machine Learning and Neural Network Approaches

Abstract:

Credit card fraud is a big concern in the financial industry -- as fraudulent transactions cause substantial financial loss to both consumers and businesses. Fraud methods have evolved in a way that has made detection very challenging, creating the need for advanced systems that can detect fraud in real-time. This project, aligned with the deep learning theme, aims to play a role in this challenge by developing and comparing various machine learning models for credit card fraud detection, with a focus on both traditional methods and neural network architectures.

I'll utilize the Credit Card Fraud Detection dataset available on Kaggle, which contains transactions made by European cardholders over two days in September 2013. This dataset presents a challenge due to its extreme class imbalance, with only 0.172% of the 284,807 transactions labeled as fraudulent. Also, the dataset includes a "time" feature that does not represent continuous or meaningful temporal sequences for individual cardholders, which complicates the use of sequence-based models like Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs). Given the nature of this dataset, these sequential models may not fully exploit the data's structure. Instead, this project focuses on methods which are likely better suited for non-sequential data.

The primary research questions this project seeks to answer are:

1. How does the performance of traditional machine learning models (such as logistic regression and random forest) compare with neural network architectures (including multi-layer perceptrons and autoencoders) in detecting credit card fraud?

2. Can we effectively handle the extreme class imbalance inherent in credit card fraud detection?
3. What insights can we gain about the indicators of fraudulent transactions, even with anonymized data?
4. Can feature selection techniques improve model accuracy and reduce computational complexity?

This project will conduct a comparative analysis of traditional machine learning models—logistic regression and random forest—alongside neural network-based architectures such as multi-layer perceptron's (MLPs) and autoencoders. While the study being compared (Mienye et al., 2024) utilizes LSTMs and GRUs, which are effective for datasets with clear temporal dependencies, this project will focus on models that are more suited to the structure of this particular dataset, where the time variable does not represent continuous sequences for individual cardholders. Instead, autoencoders are included due to their capability for anomaly detection through the learning of compressed representations, which might better capture the non-linear patterns in transaction data. Techniques to address class imbalance, including oversampling, undersampling, and synthetic data generation (SMOTE), will also be explored to improve model performance.

Results will be compared against the findings of Mienye et al. (2024), while possibly introducing new insights by focusing on models more appropriate for non-sequential data. This project will be implemented in Python, using scikit-learn for traditional machine learning models and TensorFlow/Keras for neural network models. Data manipulation and visualization will be handled using pandas, numpy, matplotlib, and seaborn.

The ultimate goal of this project is to develop an accurate and efficient credit card fraud detection system that identifies fraudulent transactions with low false-positive rates, by using models that are suited to the characteristics of the dataset.

References

Mienye, I. D., & Jere, N. (2024). Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions. *IEEE Access*, *12*, 96893–96910.

Spann, Delena D. *Fraud Analytics : Strategies and Methods for Detection and Prevention*. 1st edition. Hoboken, New Jersey: Wiley, 2014. Print.

Raschka, S., & Mirjalili, V. (2019). *Python machine learning : machine learning and deep learning with python, scikit-learn, and tensorflow 2* (Third edition.)