Credit Card Fraud Detection using LSTM and Attention

Aayush Kumar Singh (2021A7PS0430H)
Abiram M (2021A7PS0329H)
Raguram Venkatesan (2021A7PS0150H)
Under the Guidance of Aneesh Chivukula

December 18, 2024

Credit card fault detection using LSTM and Attention
(under Aneesh Chivukula)

Abstract

We were tasked with a semester-long DL project, focusing on taking on an industry-grade real-life problem. After considering a lot of different problem statements, we narrowed it down to one: "Credit Card Fault Detection." The main objective is to distinguish between fraudulent and normal transactions.

As students, we were very much interseted in knowing the inner workings of LSTM Models and attention layers for a couple of reason. First and foremost reason was to understand how DL models understand and interpret context, memory, and long term sequences. Another reason was to use topical and relevant methods to see how parallelization and consequential scalability can be achieved. So the only we could do this is by experimenting and doing a project on this

The results suggest that attention-based LSTM models can significantly enhance fraud detection, thereby reducing financial risks. More of how this was achieved would be discussed in detail in this paper.

1 Introduction / Background

1.1 Motivation and Problem Statement

Credit card fraud is a serious financial and security concern that affects consumers and businesses alike. With the rise of e-commerce and internet transactions, detecting fraud in real-time is crucial for mitigating losses and maintaining trust in financial systems. However, fraud detection is challenging due to the highly imbalanced datasets, complex patterns, and evolving fraud strategies.

1.2 Objectives

The primary objectives of this project are:

- Accurately classify data using an imbalanced dataset.
- Utilize LSTM models with an attention mechanism to better capture temporal patterns.
- Analyze the impact of different optimizers, scaling techniques, and oversampling methods on model performance.

1.3 Approach Overview

We utilized the publicly available Kaggle credit card fraud detection dataset. Data preprocessing included feature removal and scaling. Class imbalance was addressed using SMOTE and ADASYN oversampling. Baseline LSTM models were trained and compared to an LSTM model with an attention mechanism. Various optimizers, regulariztion and scaling strategies were tested, with the best-performing approach achieving approximately 95.2% F1-score, indicating high precision and recall.

2 Related Work

Earlier studies employed machine learning techniques such as logistic regression, decision trees, and random forests. Recent advancements include using deep learning models like autoencoders and LSTMs to identify complex sequential patterns. Attention mechanisms have proven effective for highlighting relevant features in sequential data. Our work extends these approaches by integrating modern techniques to create an optimal detection pipeline.

3 Dataset and Features

3.1 Description of the Data

The "creditcard.csv" dataset contains 284,807 transactions, including 492 fraudulent transactions. Features are numerical values derived from PCA transformations to ensure anonymity. The dataset also includes "Amount" and "Time" features.

3.2 Feature Engineering

We removed redundant features (e.g., V3, V5, V6, etc.) based on preliminary analysis, retaining nine features. Data scaling was performed using StandardScaler or RobustScaler.

3.3 Data Splitting

Data was split into training (70%) and testing (30%) sets after oversampling to ensure a balanced class distribution in the training set.

4 Methodology

4.1 Task Definition

The task involves binary classification: predicting whether a transaction is fraudulent (1) or legitimate (0) based on a 1x9 input vector.

4.2 Model Selection

We implemented and compared:

- A baseline LSTM model using the Adam optimizer.
- An LSTM model with an attention layer.
- Variants of LSTM models with RMSProp and SGD optimizers and different scaling techniques.

The attention mechanism was added to help the model focus on relevant time steps or features.

4.3 Hyperparameter Tuning

Experiments included tuning LSTM units, dropout rates, batch sizes, and optimizers. The area under the ROC curve was used to evaluate configurations.

4.4 Training Approach

Cross-entropy loss was used, and models were trained for 20 epochs. Regularization techniques (dropout, L2) and early stopping were employed to address overfitting.

4.5 Implementation Details

Models were implemented in Python using TensorFlow/Keras, utilizing additional libraries like scikit-learn for over-sampling and Matplotlib for visualization.

5 Experiments, Results, and Discussion

5.1 Experimental Setup

Steps included:

1. Data preprocessing and oversampling.
2. Feature scaling.
3. Training baseline and attention-based LSTM models.
4. Testing various optimizers and observing performance differences.

5.2 Evaluation Metrics

Metrics included precision, recall, F1-score and confusion matrices, with an emphasis on precision and recall for fraud detection.

5.3 Results

- **Baseline LSTM (Adam)**: Achieved about 94.34% accuracy. Precision: 96.18%, Recall: 92.3%.
- **LSTM with Attention (Adam)**: Achieved about 94.83% accuracy, Precision: 96.28%, Recall: 93.30%.
- **LSTM (RMSProp)**: F1-score around 92.75%.
- **LSTM (SGD)**: F1-score around 92.35%.
- **RobustScaler with LSTM (Adam)**: F1-score around 95.25%.
- **ADASYN oversampling with LSTM (Adam)**: F1-score around 90.91%.
- **More heavily regularized LSTM (Adam)**: F1-score around 93.52%.

5.4 Discussion

The attention-based LSTM significantly improved precision and accuracy, though recall decreased slightly. Optimizers showed minor performance variations. Scaling methods and oversampling techniques like SMOTE were robust, with ADASYN offering moderate improvements.

6 Conclusion

6.1 Summary of Findings

We developed and evaluated an LSTM-based model for credit card fraud detection, experimenting with attention mechanisms, oversampling techniques, scaling methods, and different optimizers. The attention-based LSTM model achieved the best trade-off between precision and recall, outperforming the baseline. We were able to conclude that the importance of the optimizers, regularization and scaling methods cannot be overstated in Deep Learning tasks such as this.

6.2 Contributions

This project demonstrates the benefits of attention mechanisms and oversampling for fraud detection on imbalanced datasets. The tuning of the parameters also showed significant improvements in the evaluation metrics.

6.3 Implications and Applications

Our findings can guide financial institutions in implementing effective fraud detection systems. The improved Recall and Precision can help reduce costs associated with investigating false positives.

6.4 Limitations

- Dataset reflects a single scenario and time period.
- Potential overfitting to synthetic patterns.
- Computational limitations for real-time deployment.
- Results depend on hyperparameter tuning and resource constraints.

6.5 Future Work

- Explore ensemble models or transformer-based architectures.
- Adapt to dynamic fraud patterns.
- Implement Bayesian hyperparameter optimization.
- Investigate cost-sensitive learning techniques.

7 References

1. Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, Gianluca Bontempi. "Calibrating Probability with Undersampling for Unbalanced Classification," 2015 IEEE Symposium Series on Computational Intelligence.
2. Kaggle Credit Card Fraud Dataset: `https://www.kaggle.com/mlg-ulb/creditcardfraud`
3. François Chollet et al. Keras Documentation: `https://keras.io/`
4. TensorFlow Documentation: `https://www.tensorflow.org/`

8 Appendix

8.1 Code

The code repository is available at: `https://github.com/Poseidon724/DL_Project`

(All of the necessary visualisations and the consequent inferences have been made in the ipynb file using matplotlib library. Please refer to it.)

8.2 Model Architecture Details

The basic architecture of the attention-based LSTM model includes:

- Input Layer: shape (1,9).
- Two LSTM Layers with dropout.
- Attention Layer.
- Dense Output Layer with Sigmoid Activation.

with modifications done to experiment with different parameters.