DOI: 10.21917/ijsc.2026.0567

CNN-GRU MODEL FOR CREDIT CARD FRAUD DETECTION

Srikar Ayyagari and Sai Shyam
Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning, India

Abstract

Financial fraud in online credit card transactions poses significant challenges due to its increasing prevalence and the highly imbalanced nature of transactional data. This paper proposes a hybrid deep learning framework combining one-dimensional convolutional neural networks (CNN) and gated recurrent units (GRU) to effectively capture both spatial and temporal features of transaction sequences. Bayesian optimization is employed to fine-tune the model's hyperparameters, improving detection performance without relying on synthetic oversampling. Evaluated on the widely-used European credit card fraud dataset, the proposed CNN-GRU model achieves superior results with an accuracy of 0.9996, an AUC-ROC of 0.9693, and an AUC-PR of 0.8709. These findings highlight the model's robustness in identifying rare fraudulent transactions, outperforming several state-of-the-art methods and demonstrating the practical utility of deep learning combined with Bayesian optimization in fraud detection.

Keywords:

Financial Fraud, Credit Card Transactions, Transactional Data, Convolutional Neural Network.

1. INTRODUCTION

The increasing use of the internet and online banking has led to a rise in financial fraud in online transactions. In the year 2023, around 1.13 million cases of financial fraud are reported in India alone. Financial fraud encompasses a range of illicit activities, including identity theft, unauthorized money transfers, and fraudulent incidents occurring during transactions. Credit card fraud detection is one such kind of financial fraud, wherein the card transactions can either be offline, using a physical card, or online, using card and other banking details. Such a scenario of fraud occurs when the details of the card are stolen by a malevolent entity.

Credit card fraud detection is inherently challenging due to the highly imbalanced nature of real-world transaction data, where fraudulent transactions constitute only a tiny fraction of all records. Conventional rule-based systems struggle to keep pace with evolving fraud patterns, motivating the adoption of data-driven methods, including behavioral analysis, geospatial analysis, text analysis, machine learning (ML), and deep learning (DL). Within this space, both supervised and unsupervised learning have been explored for anomaly detection, where fraudulent transactions are treated as outliers relative to the distribution of legitimate transactions. When the dataset is large, anomaly detection approaches may model the majority class using probabilistic distributions or mixture models and flag points with low likelihood as potential fraud.

Recent work has demonstrated the effectiveness of various ML and DL models for credit card fraud detection. However, many existing methods either rely heavily on oversampling, which can introduce overfitting and reduce generalization, or they focus on architectures that do not jointly exploit both local feature interactions and temporal dependencies within transaction sequences. In addition, hyperparameter tuning is often performed using manual trial and error, grid search, or random search, which can be inefficient and may leave performance gains untapped.

In this work, a hybrid deep learning model combining one dimensional convolutional neural networks (1DCNNs) and gated recurrent units (GRUs) is proposed for credit card fraud detection on the European credit card fraud dataset [13]. The CNN layers are used to extract local feature patterns from transaction attributes, while the GRU layers model temporal dependencies and sequential behavior in the data. Bayesian optimization is employed to tune key hyperparameters of the CNN-GRU architecture, reducing manual intervention and improving model performance on an extremely imbalanced dataset. The model is evaluated using metrics suitable for skewed data, including accuracy, AUCROC, AUCPR, F1score, and Matthews correlation coefficient (MCC), and it is compared against several state of theart ML and DL baselines.

1.1 OBJECTIVES

The primary objective of this study is to design a deep learning model that can effectively detect fraudulent credit card transactions in a highly imbalanced real-world setting without relying on synthetic oversampling techniques.

1.2 CONTRIBUTIONS

The main contributions of this paper are summarized as follows:

1. We propose a hybrid CNN-GRU model for credit card fraud detection, where the optimal parameters are obtained by Bayesian Optimization.
2. We compare our model to the current SOTA benchmarks, and note that this framework allows for better detection of credit card fraud.

The article is structured as follows: Section 2 provides a brief summary on the current literature in this field. Section 3 gives a brief overview of the Methodology followed in this work. We look at the dataset that is chosen, the preprocessing and tuning techniques employed in it. We also describe the various metrics considered for evaluating our model. The implementation of the model, the results obtained from it and comparison with other existing state-of-the-art models is present in section 4. A brief discussion on the results and conclusion is given in section 5.

2. LITERATURE SURVEY

2.1 TRADITIONAL ML METHODS

Early work in anomaly detection often relies on statistical modeling and classical ML techniques. Statistical approaches treat the dataset as a normal distribution, identifying anomalies as points that deviate significantly from the mean. Gaussian Mixture

Models (GMMs) are widely used to detect low-probability data points [1].

Classical machine learning techniques, including Logistic Regression, LightGBM, XGBoost, CatBoost, and various soft-voting ensemble combinations, were evaluated in [2]. A sequential ANN was also tested and found to consistently outperform these ML approaches. Metrics such as accuracy, AUC-PR (preferred for imbalanced datasets), F1-score, and MCC were used, and Bayesian hyperparameter tuning further refined performance.

In [3], SMOTE was used to address data imbalance, with Decision Tree (DT) and Random Forest (RF) models compared. RF consistently performed better due to its ability to capture complex patterns. A broader comparison across datasets in [4] showed that ML methods often perform competitively, especially on smaller datasets where deep learning tends to underperform.

2.2 DEEP LEARNING MODELS FOR ANOMALY DETECTION

Deep learning methods have been widely adopted to overcome the shortcomings of classical techniques. LSTM-based models were explored in [5]; LSTMs address the vanishing-gradient problem and capture long-term temporal dependencies. The LSTM model outperformed autoencoder-based methods and traditional ML models when evaluated using accuracy and loss.

In [6], ANN and CNN models were compared. CNN without pooling achieved the best performance, followed by CNN with pooling, and both surpassed ANN. Similarly, [7] implemented non-sequential models using 1D-CNNs, pooling, and batch normalization. Models without Max Pooling provided better results.

A hybrid model combining 1D-CNN and GRU was proposed in [8], along with the Navo Minority Over-Sampling Technique (NMOTe) for addressing imbalance. This architecture performed strongly across multiple datasets. approaches. Metrics such as accuracy, AUC-PR (preferred for imbalanced datasets), F1-score.

Deep learning models generally require large amounts of data and computational resources. They may overfit smaller datasets, and training them is significantly more expensive compared to classical methods.

2.3 FEATURE ENGINEERING AND HYBRID METHODS

Feature extraction techniques such as autoencoders enable efficient dimensionality reduction and representation learning. Autoencoders learn compressed embeddings through an encoder-decoder structure. In [9], the encoder's output was used as input features for a LightGBM model, improving performance by combining learned representations with a strong gradient-boosting framework.

Hybrid and representation-learning approaches introduce additional complexity and require careful tuning to avoid losing important information during dimensionality reduction.

Deep learning approaches are favored in this study because they inherently perform automated feature engineering and are capable of learning complex data relationships without manual intervention [10].

3. METHODOLOGY

This work frames credit card fraud detection as a binary classification problem, where an input vector of transaction features $x \in R^d$ is mapped to an output probability $p \in [0,1]$ indicating the likelihood of fraud (y=1) or no fraud (y=0). The challenge lies in detecting the minority class (fraud) amid extreme class imbalance, motivating the use of robust deep learning methods that can automatically extract relevant data representations and perform reliable classification.

3.1 DATASET DESCRIPTION

The dataset chosen to evaluate our algorithms is the European credit card fraud dataset, which contains the information of transactions made by credit cards on two days of September 2013. The dataset is highly imbalanced as out of a total of 284,807 transaction, only 492 are fraudulent transactions. The attributes of the data contain 28 columns: V1, ..., V28 which are obtained by applying Principal Component Analysis (PCA) on the original data, along with columns Amount, Time, Class denoting whether the transaction is fraud (1) or not fraud (0). The original attributes are not available due to confidentiality of information of the customers of that card.

3.2 DATA PREPROCESSING

The dataset is first imported and then all the observations containing either null values or duplicate values are removed. The data is then standardized, and the 10 most important columns that affect the Class attribute are selected. The data is then divided into training set, validation set, and testing set. The testing set will not be used till the end, where the final model is evaluated on the testing set. This is done to ensure that the testing data resembles the real-world data as closely as possible.

3.2.1 Feature Selection:

The data consists of 31 columns. One column is the Class column. There are Time, Amount columns and V1-V28, 28 columns obtained through PCA. We wanted to simplify the input data to the network and hence feature selection is done. The ANOVA F-statistic is calculated for each column with the target Class column, and the 10 columns with the highest F-scores are chosen to represent out data. There are other ways of choosing features too. From [3], we know that the Random Forest method is a good classifier of the given dataset. We can further use it to choose the best features of the data to use for our model. Instead of transforming the attributes like in PCA, we can use feature importance of a RF model to get a measure of the contribution of each feature to the predictive power of the model. The top 10 features that contribute the most can be chosen. Another way of choosing features is to measure the correlation between all features, and remove a column if it is highly correlated with another column that is not the Class column. Also retain all columns that are highly correlated with Class column.

3.3 EVALUATION METRICS

The problem is a binary classification problem. The true class can either be *Negative*(N/0) or *Positive*(P/1). And the model can also classify the observation as N or P. We classify each of the

observations as TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). The main metrics that are considered are:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{F1-score} = \frac{2(\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \quad (4)$$

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{TN + FP} \quad (6)$$

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n}\left[-y_i \log(p_i) - (1 - y_i)\log(1 - p_i)\right] \quad (7)$$

which is the binary cross entropy loss function. We are also considering Mathews Correlation Coefficient, AUC-ROC: Area under the ROC curve, AUC-PR: Area under the Precision-Recall curve.

The generally used performance measure for fraud-detection problems is AUC for ROC curve, whose value can be seen as a probability that the classifier ranks frauds higher than genuine transactions and average precision, or the AUC for PR curve is also a generally used metric for fraud-detection [11].

3.4 MODEL ARCHITECTURE

The deep learning model put forward by [5], consisting of LSTM layers was first implemented. Their model was first enhanced by changing the number of layers and even tuning the hyperparameters further. But there has not been any significant improvement in the results.

Then the CNN-GRU model of [8] was considered for implementation. Oversampling techniques such as SMOTE etc. have many drawbacks in terms of Overfitting, less generalization to new data, loss of information and reduced model interpretability. These lead us to consider the given model without using their oversampling technique. The model was further enhanced.

The results from [7] and [4] convey that it is better to remove any pooling layer after a convolution layer in this dataset, as it leads to loss of information and the model usually performs better without pooling layers. We added batch normalization layers to reduce overfitting by the model. The number of neurons in each layer was changed. Following the approach in [2], Bayesian-based Hyperparameter tuning to get the optimum number of neurons in each layer of the model. Bayesian Optimization is chosen since it is generally better than random search or grid search and can even be better than manual expert optimization [12]. The components of our model are shown in Fig.1.

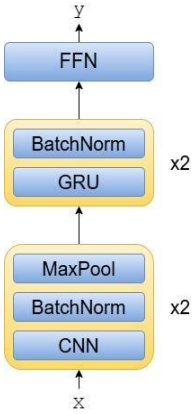

Fig.1. CNN-GRU Model Architecture

3.4.1 Hyperparameter Tuning:

We have also experimented with both L1 and L2 regularization methods. Regularization methods are used mainly to reduce overfitting of the model by modifying the loss function of the model by adding a regularization term to it . The regularization term will be higher for complex models and lower for simpler models. Hence, the model will be incentivized to reduce its complexity and thus leads to reduction in overfitting.

But since dropout and batch normalization layers are present in the model, there is no need for any other method to reduce overfitting. This is also reflected in the results when it is found that the performance of the model did not increase with regularization. In fact, it deteriorated slightly since the model is not able to capture the data fully due to increase in variance. So, the regularization term is dropped.

The dropout rate after each layer is also experimented upon. Too high dropout leads to loss of data, and too low dropout is almost similar to zero dropout and there is no effect of it. Bayesian based hyperparameter tuning is also tried for dropouts but it was found that the search space became too big for it and the processing time is increasing exponentially.

Hence, through trail and error, a dropout rate of 0.15-0.2 is found optimal. Although LSTMs are better at capturing long-term dependencies, as mentioned in [5], an attempt was made to replace GRU cells with LSTM cells, but there wasn't a significant change in the results. Therefore, GRUs were retained since they are computationally less expensive than LSTMs.

4. RESULTS

The results of our CNN-GRU model was compared with various other state-of-the-art models and was tabulated below.

Table.1. Performance Evaluation of Various Models

Model	Acc.	Rec.	Prec.	F1-Score	MCC	AUC-ROC	AUC-PR	Loss
ANN model [2]	0.9994	0.8222	0.8043	0.8132	0.8129	0.9401	0.7922	-
LSTM model [3]	0.9996	0.7474*	0.8765*	0.8068*	-	0.9328*	-	**0.0021**
CNN model [4]	0.9585	-	-	0.8372	-	-	-	0.00392
CNN model [6]	0.999	0.775	**0.932**	0.8462	-	0.929	0.816	0.004
ML approach - LGBM	0.9991	0.799	0.7534	0.7699	0.7727	0.9472	0.7657	-
CNN-GRU Approach	**0.9996**	**0.8235**	0.918	**0.8682**	**0.8693**	**0.9693**	**0.8709**	0.0025

Note: ()* - Indicates that the value is obtained by our implementation of the model

The results show that the proposed CNN-GRU model achieves superior performance compared to existing neural network and machine learning baselines for credit card fraud detection. In particular, our approach yields the highest accuracy (0.9996), recall (0.8235), F1-score (0.8682), and MCC (0.8693) among all models evaluated. Its AUC-ROC (0.9693) and AUC-PR (0.8709) are also the highest, indicating strong discriminative power even in the presence of severe class imbalance. These results demonstrate that combining convolutional and recurrent layers, together with principled hyperparameter optimization, enables more reliable identification of rare fraud cases while maintaining low loss and balanced precision-recall trade-offs.

From our test data, the Confusion matrix came to be:

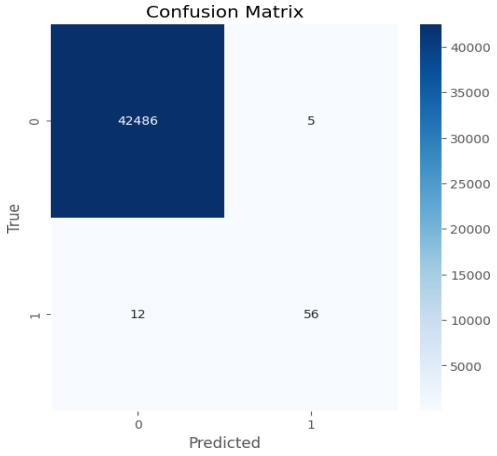

Fig.2. Confusion matrix

It can be seen that there are only 17 instances of mis-classification out of a total of 42559 test values. This is the reason for the very high accuracy. However, precision and recall are not extremely high. This is due to the fact that there are only 68 observations which are of positive class(Fraud), and out of them, 56 are classified correctly. Further, 5 observations that are actually not fraud, are mis-classified as fraudulent transactions.

Varying the threshold from 0 to 1, we get the ROC curve, which plots TPR(True Positive Rate) vs FPR(False Positive Rate), and also the PR curve, which plots Precision vs Recall

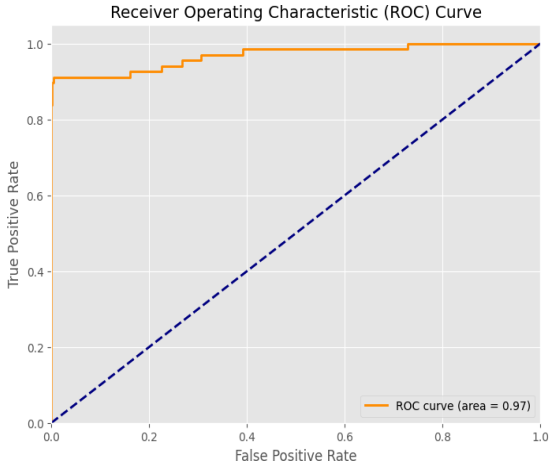

Fig.3. ROC Curve

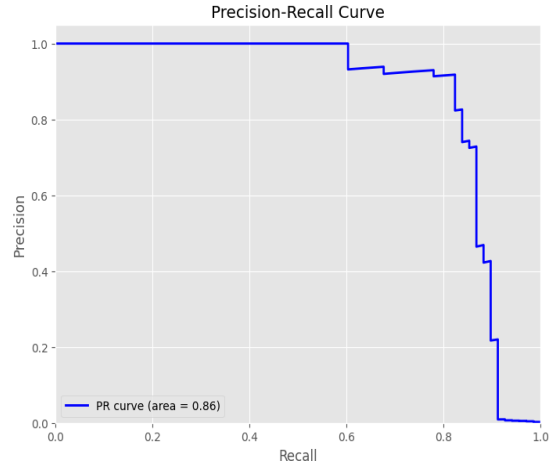

Fig.4. Precision-Recall Curve

5. DISCUSSION AND CONCLUSION

The proposed hybrid CNN-GRU model leverages one-dimensional convolutional layers to effectively extract localized feature patterns from tabular credit card transaction data, a less common yet powerful approach for this data type. These convolutional layers automatically learn relevant feature representations that may not be apparent through manual feature engineering.

The GRU layers complement this by capturing temporal dependencies and sequential behaviours inherent in transaction data, addressing the challenges posed by the highly imbalanced nature of fraudulent transaction detection.

Our results demonstrate that this combined architecture, optimized through Bayesian hyperparameter tuning, achieves superior performance on multiple evaluation metrics, including accuracy, recall, F1-score, and AUC-ROC, compared to contemporary state-of-the-art models.

Bayesian optimization proved efficient in exploring the hyperparameter space, improving performance while avoiding exhaustive manual tuning. Notably, the model performs well without oversampling techniques, mitigating risks of overfitting and preserving the integrity of the dataset distribution.

Despite these strengths, some limitations remain. The model's precision and recall, while competitive, indicate the inherent difficulty of detecting rare fraud cases with limited positive samples.

The results are currently validated on a single widely used dataset, which raises questions about generalizability to other transaction environments with different fraud patterns and feature distributions. Moreover, the black-box nature of deep learning models may hinder interpretability, which is critical for trust and regulatory compliance in financial systems.

6. FUTURE WORK

We have only looked at a single dataset until now. This is because the European credit card dataset is one of the most extensively worked upon dataset in the field. Furthermore, obtaining legitimate financial fraud datasets is challenging due to the inclusion of confidential and personal customer information. These methods can be extended to other fields to test their dataset independence and evaluate their applicability in real-world scenarios.

Feature Selection is performed by studying the significance of difference of means among different features using the F-statistic. The effects on the results if we use a different approach to process data and select features can be explored.

REFERENCES

[1] M. Bahrololum and M. Khaleghi, “Anomaly Intrusion Detection System using Gaussian Mixture Model”, *Proceedings of International Conference on Convergence and Hybrid Information Technology*, pp. 1-8, 2019.

[2] M.A.G. Hashemi, “Fraud Detection in Banking Data by Machine Learning Techniques”, *IEEE Access*, Vol. 11, pp. 3034-3043, 2022.

[3] M.A.G. Teja, “A Research Paper on Credit Card Fraud Detection”, *International Research Journal of Engineering and Technology*, Vol. 9, No. 3, pp. 1178-1181, 2022.

[4] M.A.K. Alarfaj, “Credit Card Fraud Detection using State-of-the-Art Machine Learning and Deep Learning Algorithms”, *IEEE Access*, Vol. 10, pp. 39700-39715, 2022.

[5] A.A.R. Alghofaili, “A Financial Fraud Detection Model based on LSTM Deep Learning Technique”, *Journal of Applied Security Research*, Vol. 15, No. 4, pp. 1-8, 2020.

[6] B.A. Aljohani, “Credit-Card Fraud Detection System using Neural Networks”, *The International Arab Journal of Information Technology*, Vol. 20, No. 2, pp. 234-241, 2023.

[7] B.A. Pratap, “Credit Card Fraud Detection using Deep Learning”, *Proceedings of International Conference for Convergence in Technology*, Vol. 2, pp. 1-9, 2020.

[8] K.A. Senthilselvi, “An Integration of Deep Learning Model with Navo Minority Over-Sampling Technique to Detect the Frauds in Credit Cards”, *Multimedia Tools and Applications*, Vol. 82, No. 1, pp. 1-6, 2023.

[9] H. Du, G. An, L. Li and H. Wang, “AutoEncoder and LightGBM for Credit Card Fraud Detection Problems”, *Symmetry*, Vol. 15, No. 4, pp. 1-7, 2023.

[10] S.F. Ahmed, “Deep Learning Modelling Techniques: Current Progress, Applications, Advantages and Challenges”, *Artificial Intelligence Review*, Vol. 56, pp. 13521-13617, 2023.

[11] A.P. Dal, G. Boracchi, O. Caelen and C. Alippi, “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy”, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 8, pp. 3784-3797, 2018.

[12] J. Snoek, L. Hugo and R.P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms”, *Advances in Neural Information Processing Systems*, Vol. 3, pp. 1-12, 2012.