JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DEEP LEARNING AND SIMULATION FRAMEWORK TO IMPROVE CREDIT CARD FRAUD DETECTION

[1]Raghad H. Almakrami 1st, [2]Lama Hamad 2nd , [3]Ahmad M. Almosabi 3rd

1 Department of Computer Science, College of Computer Science, Najran University, Najran, Saudi Arabia 1st

2 Department of Computer Science, College of Computer Science, Najran University, Najran, Saudi Arabia 2nd

3 Department of Computer Science, College of Computer Science, Najran University, Najran, Saudi Arabia 3rd

[1]Email:443302865@nu.edu.sa 1st,[2]Email: 443300104@nu.edu.sa 2nd ,Email:amalkheder@nu.edu.sa 3rd

Abstract : This study focuses on the development of an effective deep neural network (DNN) model to detect fraudulent financial transactions. The data set was previously processed by using the standard measurement and divided into training and testing groups to ensure a fair evaluation of the model. A dense three-layer neural network was trained using the binary_cross entropy loss function and the binary taxonomy X-sin activation function. The model achieved excellent results, with a training accuracy of 99.96% and minimal differences in loss between the training and testing stages, indicating its durability and ability to generalize effectively,The model was further tested by simulation, where it succeeded in identifying a new transaction with a fraud probability of only 0.0002% ,These results highlight the
reliability of the form and its willingness to deploy in the real world in fraud detection systems.

Keywords: *Financial Fraud Detection ; Deep Neural Networks (DNN) ; Discrete-Event Simulation ; Imbalanced Data Classification ; F1-Score*

I.INTRODUCTION

In recent years, financial fraud has increased significantly, posing a serious risk to the stability of the global economy and the trust in financial institutions. The rapid growth of digital financial transactions, especially during and after the COVID-19 pandemic, has led to more complex fraudulent techniques [1]. As financial systems rely more on electronic transactions and automated processes, the vulnerability to fraud has grown, impacting both individuals and institutions. Finding financial fraud is difficult because of the vast amounts of data and the constantly changing tactics of fraudsters. Traditional systems that use fixed rules cannot effectively identify modern fraudulent patterns. Therefore, machine learning has become essential for detecting anomalies and suspicious activities in financial statements. Some of the top models used in this area include Logistic Regression, Random Forest, and Deep Learning [6]. These models can manage the complex relationships in data and accurately predict fraudulent behavior. However, researchers face a major challenge: there is a lack of public financial databases for research, due to privacy and confidentiality rules. To tackle this issue, simulated environments can create artificial financial data that mimics real-world situations while protecting user privacy [2],[3]. This project uses the Python SimPy environment to model the flow of financial transactions and create various scam scenarios[4],[9]. The data generated will help train and evaluate machine learning models, assessing their performance with metrics like Accuracy, Precision, Recall, and F1-Score [7],[8]. The goal of this research is to create a system for simulating and detecting financial fraud by combining simulation and machine learning techniques. Specifically, the SimPy environment will generate data, and a Deep Learning model will perform multi-category classification. This approach provides a flexible and safe way to evaluate model performance and improve accuracy in realistic financial situations

In order to detect financial fraud using machine learning algorithms, an understanding of the basic tools and concepts that support model construction and evaluation is offered.

1-Deep Neural Networks (DNNs) The “deep” in deep nets refers to the presence of multiple hidden layers that enable the network to learn complex representations from input data. These hidden layers enable DNNs to solve complex ML tasks; more “shallow” artificial networks cannot handle [6]. Hidden layers in a DNN are dense (fully connected) layers. Each neuron in a dense layer is connected to every neuron in the previous and subsequent layer, which makes DNNs highly suitable for learning complex relationships in data. such as financial fraud detection. In this research, the Karas and TensorFlow frameworks were used to build and train the DNN model in a practical and efficient way, due to the strong possibilities they offer in the implementation of deep learning algorithms and improve their performance using acceleration via the GPU.

2-Pre-processing of data: normalization and coding the pre-processing phase is an essential step in preparing the data before entering it into the form.

II. RESEARCH METHODOLOGY

The StandardScaler normalization technique was used to ensure that all characteristics have the same numerical range, preventing any variable from overly affecting the learning process due to different scales [5]. The One-Hot Encoding method has also been applied to categorical variables containing seven classes, to convert them into binary vectors used in the final output layer of the model, allowing accurate and balanced classification.

Procedures (Implementation of Methodology) To provide a fair evaluation, the dataset was divided using the train test split function. Eighty percent was set out for training, with twenty percent reserved for testing. The broker used stratify=y to maintain an unbalanced distribution of Fraud Class 1 across both groups. Following the split, StandardScaler was used to standardize the features [5]. The fitting statistics were derived from the training data (\mathbf{X\ train}), and the same parameters were applied to the test data (\mathbf{X\ test}) to prevent information leaking. A deep neural network (DNN) was constructed with three dense layers [6]. The binary_cross entropy loss function and the sigmoid activation function were utilized in the output layer to solve the binary classification issue.

Training and Evaluation Outcome

The training phase lasted ten epochs and demonstrated consistent and effective performance. It finished with a final loss of 0.0017 on the training set and an amazing accuracy of 0.9996%.

Evaluation measures in multiple classification

1.Accuracy (Accuracy)

accuracy is the percentage of correct classifications that a trained machine learning model achieves, the number of correct predictions،The validity of the model is generally.

2.Specific accuracy (Precision)

indicator of a machine learning model’s performance – the quality of a positive prediction made by the model Precision refers to the number of true positives (fraudulent) among all cases that the model is expected to be fraudulent.

3.Recall (Recall) Measures the model’s ability to find all the positive instances. the extent to which the model “covers” the correct cases.

4.F1-Score The harmonic mean of precision and recall. It balances the two metrics into a single number, making it especially useful when precision and recall are in trade-off [7],[8]. This indicator is very important in detecting fraud, because it combines the model's ability to avoid false alarms (Precision) and its ability to detect all fraudulent cases (Recall).

5.Confusion matrix (confusion matrix) Confusion matrix is a simple table used to measure how well a classification model is performing. It compares the predictions made by the model with the actual results and shows where the model was right or wrong. These metrics were calculated using the macro average to evaluate the efficiency of the model across all categories in a balanced way, even in cases of data imbalance،These metrics help provide a comprehensive understanding of the model's ability to distinguish between sound and fraudulent financial transactions.

After confirming The model is ready for usage by describing the methods for successfully saving, uploading, and simulating a new payment transaction. Model Storage and Access: After confirming high performance indicators, the model was saved for future usage without retraining. The model.save() function was used to save the model in the desired format (.keras). This stage preserves both the network's weights and its ultimate structure.

Simulation of a New Transaction: A test of a single unnoticed push transaction was carried out to assess the model's ability to classify in a real-world setting. The incoming transaction data was processed using the standardScaler that matches the trained standard. This step is crucial for ensuring the accuracy of the predictions.

The application of simulation techniques, as utilized here, is vital for testing the system under controlled conditions before live deployment [9], [10].

In summary, the dataset was produced, a DNN model was built and trained, different performance measures were used to evaluate it, and the effectiveness was tested on simulated transactions. This methodology ensures that the model generalizes well and accurately detects fraudulent transactions even when the data is skewed.

III. Results and Discussion

1. Outlining preliminary results

Computational efficiency: Training the model over ten epochs was successful. Each training step took about 2 ms, showing high processing efficiency. Accuracy of training versus loss: The training accuracy started at 0.9980 in the first epoch and rose to 0.9996 in the last epoch. The model's effective learning was also shown by the loss function value, which fell sharply and steadily from 0.0138
to a low of 0.0017.

2. Evaluation of Test Group Performance
Key performance indicators from hidden data (\mathbf{X\ test} and \mathbf{y\ test}) are outlined in this section. The tables below provide an evaluation of the model's effectiveness and how well it applies to new data.

A. Training vs Testing Performance

The model’s generalization ability was evaluated by comparing its performance on the training set versus the unseen test set. Table 1 presents the final accuracy and loss for both sets.

Table 1: Comparison of Training and Testing Performance

Metric	Final Training value	Final Testing Value	Conclusion from comparison
Accuracy	0.9996	0.999473.	Highly consistent performance
Loos	0.0017	0.0032.	Values are closely aligned

The table shows that the model generalizes well with minimal differences between training and testing, indicating a low risk of overfitting.

B. Key Classification Metrics on Test Set
The model’s ability to detect fraud in unseen data was evaluated using classification metrics, which are especially important for imbalanced datasets.

Table 2: Key Classification Metrics on Test Set

Metric	Value	Interpretation
Overall	99.9473%	The percentage of correct predictions across all classes
Test Loss	0.0032	The final error measure of the model on the unseen data
Precision	0.90	The ability of the model to avoid false positives (minimizing wrongly flagging a normal transaction as fraud)
Recall	0.85	The model's ability to detect all actual fraudulent cases (minimizing missed fraud
F1-Score	0.87	The harmonic mean of Precision and Recall, the most critical metric for imbalanced

The high F1-Score indicates the model reliably detects fraudulent transactions even when they are rare. The close alignment of accuracy and loss between training and testing demonstrates robust generalization.

C. Simulation and Practical Application

The model was further tested with a simulated new transaction to evaluate real-world applicability. The probability of fraud was extremely low, and the transaction was correctly classified as normal.

Table 3:Simulation Prediction Outcome

Metric	Value	Interpretation
Fraud	0.0002%	The calculated probability of the transaction being fraudulent,which is extremely low
Final Classification	0	The final decision is a normal transaction (No fraud), based on the 0.5 threshold

These results confirm that the model is operationally ready and can effectively detect fraud in practical scenarios.

D.Overall Discussion

•The close match between training and testing metrics demonstrates that the model generalizes well and avoids overfitting.

•High Precision and Recall indicate that the model both avoids false alarms and detects actual fraud effectively.

•Simulation results show that the model can handle unseen transactions accurately, proving readiness for deployment in financial systems.

•Overall, the results validate that the proposed DNN model is suitable for real-world financial fraud detection, even in cases of imbalanced data.

IV.Applications and Simulation

This section demonstrates that the model is ready for usage by describing the methods for successfully saving, uploading, and simulating a new payment transaction. Model Storage and Access: After confirming high performance indicators, the model was saved for future usage without retraining. The model.save() function was used to save the model in the desired format (.keras). This stage preserves both the network's weights and its ultimate structure.

Simulation of a New Transaction: A test of a single unnoticed push transaction was carried out to assess the model's ability to classify in a real-world setting. The incoming transaction data was processed using the standardScaler that matches the trained standard. This step is crucial for ensuring the accuracy of the predictions.

The application of simulation techniques, as utilized here, is vital for testing the system under controlled conditions before live deployment [9], [10].

In summary,The simulation results confirm that the developed deep neural network model is fully functional, loaded. efficiently, and is able to classify new transactions with high confidence. The very low probability of fraud (0.0002\%) and the final rating of the number '0' prove the readiness of the system for practical and actual application in fraud detection environments.

V.*Conclusion*

The deep-port neural network model demonstrated exceptional performance and high accuracy in distinguishing between legitimate and fraudulent transactions.Close alignment between training and testing results confirmed the strong generalization ability of the model and reduced the risk of over suitability. Moreover, a new invisible transaction simulation verified the operational readiness of the model, achieving a prediction of the possibility of almost zero fraud. Thus, the developed system can be used confidently in real-world financial environments to enhance transaction security, reduce losses and support automated fraud detection efficiently and accurately.

References

[1] X. Zhu, X. Ao, Z. Qin, Y. Chang, Y. Liu, Q. He, and J. Li, "Intelligent financial fraud detection practices in post-pandemic era," *XINN*, pp. 1–12, 2021. doi: 10.1016/j.xinn.2021.100176

[2] M. Kraus and S. Feuerriegel, "Decision support from financial disclosures with deep neural networks and transfer learning," *arXiv preprint arXiv:1710.03954*, Oct. 12, 2017, pp. 1–34.

[3] E. A. Lopez-Rojas and S. Axelsson, "Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research," in *Advances in Computational Social Science and Social Simulation*, M. Miguel, M. Amblard, J. Barceló, and F. Madella, Eds., Barcelona: Autònoma University of Barcelona, 2014, pp. 1–10.

[4] E. A. Lopez-Rojas and S. Axelsson, "A review of computer simulation for fraud detection research in financial datasets," in *2016 Future Technologies Conference (FTC)*, pp. 932–935, 2016.

[5] N. Matloff, "Introduction to Discrete-Event Simulation and the SimPy Language," Feb. 13, 2008, pp. 1–33.

[6] R. Cont, "Scaling and correlation in financial data," *arXiv preprint cond-mat/9705075*, 1997.

[7] M. Owusu-Adjei, J. B. Hayfron-Acquah, T. Frimpong, and G. Abdul-Salaam, "Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems," *PLOS Digital Health*, vol. 2, no. 11, e0000290, 2023.

[8] J. Miao and W. Zhu, "Precision-Recall Curve (PRC) Classification Trees," Nov. 17, 2020, pp. 1–19.

[9] E. A. Lopez-Rojas, "Applying simulation to the problem of detecting financial fraud," Blekinge Tekniska Högskola, 2016.

[10] E. A. Lopez-Rojas, S. Axelsson, and D. Baca, "Analysis of fraud controls using the PaySim financial simulator," *International Journal of Simulation and Process Modelling*, vol. 13, no. 4, pp. 377–386, 2018.