FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 5, May, 2025, pp 263 - 270
DOI: https://doi.org/10.33003/fjs-2025-0905-3379

AN ENSEMBLE OF UNSUPERVISED DEEP LEARNING MODELS FOR CREDIT CARD FRAUD DETECTION

Aguguo Ausbeth Chiemeka and *Nurudeen M. Ibrahim

Nile University of Nigeria, Jabi, Abuja

*Corresponding authors' email: nurudeenimahmud@gmail.com

ABSTRACT

As the use of credit cards for transactions increases, so do frauds too. Global networking offers criminals just as many new opportunities as it do for ordinary users. The good news is that technology for detecting and preventing credit card fraud is likewise getting better with time. Machine Learning models have been incorporated in this field to reduce the cost and time it takes in analyzing credit card transactions and detect the fraudulent ones. However, current machine learning models primarily rely on supervised learning, which requires labeled data and struggles with new fraud patterns and class imbalances. Despite this, unsupervised machine learning models tend to perform worse than the supervised learning models in detecting credit card fraud, which makes them less likely to be a go-to option in this field. This research proposes an ensemble of unsupervised deep learning models, specifically Deep Auto encoders (AEs) and Recurrent Neural Networks (RNNs), to improve the performance of unsupervised models for credit card fraud detection. This research employs a quantitative methodology, utilizing secondary data from credit card transactions. The methodology involves preprocessing the data, followed by training and evaluating the models. The individual models achieved AUC scores of 0.96 and 0.82 respectively, while the ensemble model from their combination achieved an AUC score of 0.95.

Keywords: Auto encoders, Credit card fraud, Deep learning, Ensemble Learning, Recurrent Neural Networks

INTRODUCTION

In the rapidly evolving landscape of financial transactions, credit card fraud has become a significant challenge that reflects the darker side of technological advancements in the banking industry. Tracing its roots back to the early days of credit cards in the 1950s, fraud has always accompanied technological innovations in this field. Initially, fraudsters exploited simple strategies such as forging signatures and stealing physical cards. However, as the financial industry transitioned into the digital age, fraudsters developed more sophisticated methods that used technology to exploit vulnerabilities in online transactions and payment systems to obtain an individual's credit card information. Globally, credit card fraud is a persistent issue, and some nations have high reported areas of credit card fraud even if their credit card usage is low or average (Oghenekaro & Ugwu, 2016). The evolution of credit card fraud has necessitated the development of advanced detection and prevention techniques, which makes the study of credit card fraud detection not just relevant but also crucial for maintaining the integrity of financial transactions in the 21st century.

Fraudulent transactions can manifest themselves in variety of ways, including transactions that are noticeably different from the user's typical spending patterns, exceptionally big transactions, or transactions performed at odd times or locations. They can be challenging to identify using traditional techniques for detecting credit card fraud, such auditing, in which a qualified individual manually reviews records or transactions to look for fraudulent activity (West et al., n.d.), as there are limitations a human who is doing the detection manually face, such as the speed at which fraudulent activity can be detected. Recently, machine learning techniques have proven to overcome several limitations traditional techniques suffer from, and, in this case, protect against financial losses due to credit card fraud with better results. These machine learning techniques are categorized into two types, namely supervised learning and unsupervised learning.

Supervised learning uses labeled data to train a model to make predictions or categorize new data based on the labeled data provided. Using this method to tackle credit card fraud depends on the set of previous transactions made by the cardholder or by a fraudster for which the label of the transaction (normal or fraudulent) is known. It uses this information to train a fraud prediction model to classify any new transaction as normal or fraudulent (Carcillo et al., n.d.). Unsupervised learning deals with datasets that lack explicit output labels. It aims to find patterns or relationships in the data without using predefined labels. In tackling credit card fraud, this technique aims to characterize the data distribution of transactions without relying on the knowledge of the label of transactions. They rely on the assumption that patterns that are significantly different from the normal transaction distribution are fraudulent. One of the advantages of this over supervised technique is that it can be used to adapt to changing patterns of fraud over time since they do not rely on past labeled transactions (Carcillo et al., n.d.).

A neural network is a type of Artificial Intelligence (AI) algorithm hhhthat is modeled after the human brain, aims to simulate its activity, and enable computers to process data through this simulation. Layers of connected nodes or neurons make up neural network's input, hidden and output layers. Deep Learning is a subset of machine learning, featuring neural networks with numerous hidden layers (Zhang et al., n.d.). It is a popular method to detect outliers for credit card fraud as it is particularly well-suited in analyzing complex patterns in large datasets and can learn from experience to improve accuracy over time. Deep Learning uses a variety of techniques, namely Autoencoders (AEs), Recurrent Neural Networks (RNNs), Restricted Boltzmann Machine (RBM), Convolution Neural Networks (CNNs), etc. Deep Learning models can be integrated into a singular framework to obtain a stronger model that outperforms their individual models using an approach known as Ensemble Learning (Mohammed & Kora, 2023).

Recently, several machine learning and deep learning approaches have been put in place to tackle credit card fraud

as a more accurate and efficient solution. However, majority of these techniques are trained under supervised learning. There is a drawback with this because credit card transactions provide an unbalanced set of transactions, therefore it is typically difficult to collect efficient labeled data in this field. This is because fraudulent transactions occur far less frequently than legitimate transactions. Another issue is that it can take some time to label new transaction data, which would postpone updating the supervised model (Niu et al., n.d.). Unsupervised models do not suffer from these issues as they do not rely on labeled data and can be easily modified to detect new kinds of fraud, since it does not rely on priori transaction information as well. This research aims to improve the performance of unsupervised models in credit card fraud detection, by performing an ensemble on two unsupervised models, specifically Autoencoders (AEs) and a Recurrent Neural Networks (RNN) trained without labels.

Islam et al. (2023) offers a detailed analysis of using ensemble learning for credit card fraud detection. Their work focuses on the challenges posed by imbalanced and overlapping class samples in this data. The authors propose a model called Credit Card Anomaly Detection (CCAD), which uses a stacked ensemble learning approach that incorporates both outlier detection algorithms and the Extreme Gradient Boost algorithm as a meta-learner. The paper demonstrates that their model outperforms existing approaches, especially in identifying anomalies in minority class instances.

Saraf and Phakatkar (2022) presents a comprehensive approach for detecting credit card fraud using machine learning techniques. Their study proposes a supervised hybrid ensemble model for classifying fraudulent and normal transactions by combining Random Forest and AdaBoost. They also addressed the issue of imbalanced datasets common in this field by employing oversampling methods. It achieved 98.27% area under the curve score. The authors also highlighted the benefits of an ensemble model as the individual Random Forest and AdaBoost models both achieved F1 score of 0.95% but achieved 0.97% when combined in an ensemble model.

Boucher (2020) focuses on evaluating outlier detection techniques for detecting fraudulent banking transactions. He tested two naïve methods: clustering and statistical methods, and four machine learning methods: isolation forest, local outlier factor, support vector machine, and logistic regression on three different datasets, which include a financial banking set, a credit card set, and a company audit set. From the results, logistic regression performed better than other methods with an accuracy and F1 score of 0.95%. It emphasized how relevant logistic regression is for fraud detection, especially for smaller, balanced datasets, while also providing acceptable results for other data sets. The research contributes to understanding the applicability and effectiveness of various outlier detection techniques in different data conditions.

Bodepudi (2021) compares three unsupervised algorithms: Isolation Forest, Local Outlier Factor, and One-Class SVM for fraud detection in credit card transactions. The study finds that Isolation Forest outperforms the other algorithms in terms of accuracy; it highlighted the benefits of unsupervised algorithms in detecting anomalies and fraudulent transactions, particularly in scenarios lacking labeled data.

Pumsirirat and Yan (2018) used Tensor flow library from Google to implement a credit card fraud detection system that focuses on deep learning models: Auto-Encoders (AEs) and restricted Boltzmann machines (RBMs). The authors highlighted the growing significance of fraud detection in the financial industry and the limitations of traditional rule-based methods. The study also introduced the concept of deep learning and explains how it can be used to detect fraudulent transactions. The authors highlight the effectiveness of AEs and RBMs and emphasized their ability to reconstruct normal transactions and detect anomalies. The authors conducted experiments on a dataset of credit card transactions and used AUC, precision, recall, and F1-score metrics to evaluate the performance of their models. The results show that both AEs and RBMs produce high AUC scores in detecting fraudulent transactions with large datasets having 0.96 and 0.95 respectively. The outcome of this research was to show the strengths of deep learning techniques in identifying complex fraudulent patterns in large datasets.

Rezapour (2019) presented a study that describes three specific unsupervised methods: one-class SVM, auto encoder, and Robust Mahalanobis, providing an overview of their implementation and challenges. The author emphasizes the necessity of adapting unsupervised detection models to keep pace with evolving fraudulent behaviors and the importance of considering both global and local outliers in future research. The models were evaluated using a confusion matrix to determine the precision, recall and f1-score of each model. The study concludes by suggesting that more nuanced models that incorporate cardholder behavior and transaction history may lead to improved detection accuracy. Similarly, Hadiza et al. (2025) proposed a credit card fraud detection method addressing data imbalance and poor feature selection. The study used a wrapper-based feature selection and a hybrid sampling approach combining SMOTE, random oversampling, and under-sampling. Using classifiers like KNN, Random Forest, and SVM, the method achieved improved accuracy and perfect classification on balanced data.

Credit card fraud has evolved from simple physical theft to sophisticated digital attacks, posing major risks to financial institutions. Traditional detection methods struggle with speed and accuracy, while machine learning, especially unsupervised models, offers promising alternatives despite challenges like data imbalance. This research proposes an ensemble of unsupervised deep learning models, specifically Deep Auto encoders (AEs) and Recurrent Neural Networks (RNNs), to enhance fraud detection. By avoiding the limitations of labeled data, the model adapts better to evolving fraudulent patterns. The ensemble approach leverages the strengths of both models to improve detection accuracy.

MATERIALS AND METHODS

This research developed an ensemble model for fraud detection using secondary data from a Kaggle dataset of European credit card transactions. The data, contains 284,807 transactions with 492 fraudulent ones, was preprocessed through data cleaning, normalization, and feature scaling to prepare it for training. Two models, an auto encoder and a Recurrent Neural Network (RNN) as shown in Figure 1 were selected for their abilities to detect outliers and handle sequential data, respectively. The auto encoder was trained on randomized data to focus on reconstruction errors, while the RNN was trained on sequential data to capture temporal dependencies. Both models were trained with the Adam optimizer and Mean Squared Error (MSE) loss function, and Early Stopping was used to prevent over fitting. The ensemble learning approach combined the outputs of these models using max, average, and weighted average voting methods to enhance detection accuracy. The ensemble strategy aimed to leverage the strengths of both models to achieve a more robust and reliable prediction of fraudulent transactions.

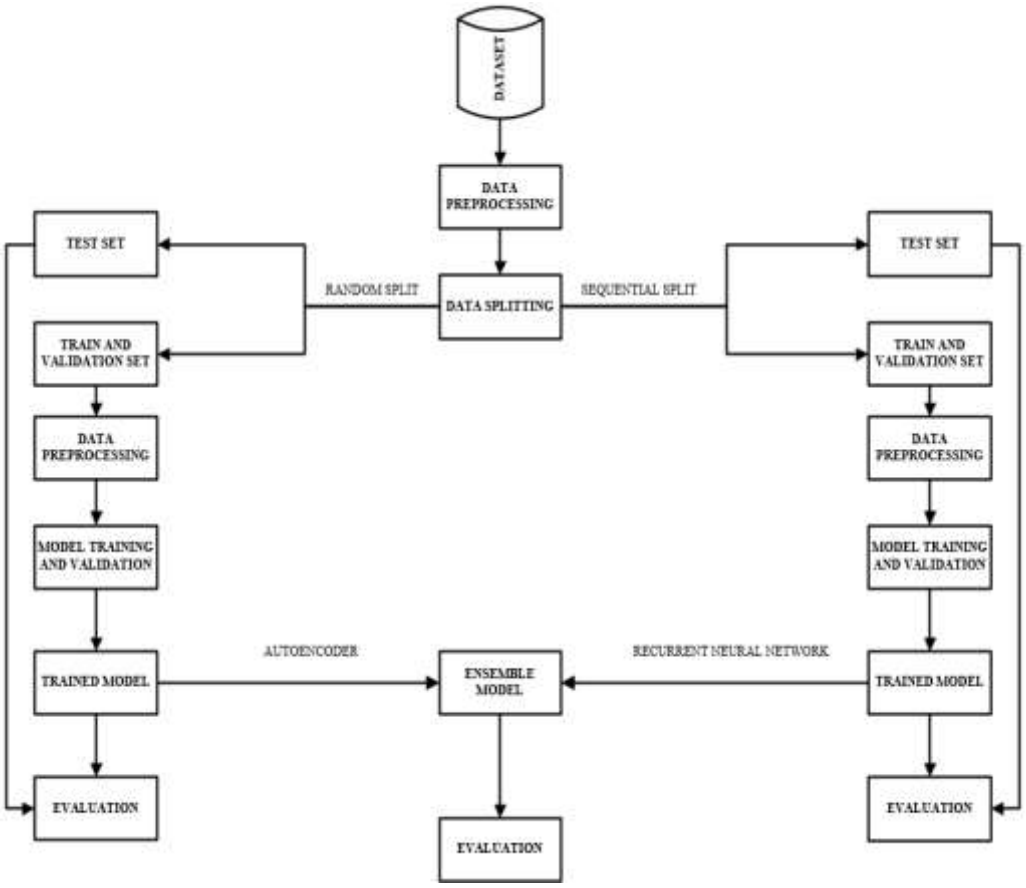

Figure 1: Research Framework

Data Collection

The type of data used for this research was secondary data sourced from Kaggle. It is a tabular dataset that contains credit card transactions made by European cardholders over a period of two days in September 2013. The dataset contains a mix of fraudulent and non-fraudulent transactions, and includes features such as transaction amount, time, and V1-V28, which are the result of a Principal Component Analysis (PCA) transformation to protect sensitive information. This dataset contains a total number of 284,807 transactions with 492 being fraudulent transactions and 284,315 being normal transactions as depicted in Figure 2. This indicates that fraudulent transactions in this dataset account for only 0.172% of all transactions, which correlates to how rare fraudulent transactions occur in a real-world setting. There were no missing values in this dataset.

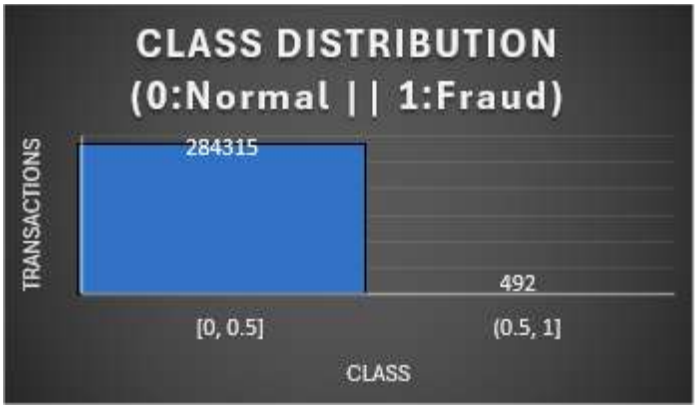

Figure 2: Class Distribution Histogram

Data Preprocessing

Data preprocessing was crucial to prepare the dataset for analysis. The first step was data cleaning, where duplicate rows were removed to ensure accurate analysis based on unique data points. The skewness of the Time and Amount features was then calculated to assess asymmetry, revealing that the Amount feature was highly skewed. A Yeo-Johnson transformation was applied to the Amount feature, normalizing its skewness from 16.98 to 0.018, making it more balanced. The dataset was split into training, validation, and test sets with a ratio of 60:40, ensuring a substantial number of fraudulent transactions in the validation and test sets. The training set contained only normal transactions, while the validation and test sets included both normal and fraudulent transactions. The data split was performed twice: a random split for the auto encoder and a sequential split for the RNN

to maintain the chronological order of transactions. Feature scaling was performed using z-score normalization to bring the Time and Amount features to a similar range as other features, preventing data leakage by fitting the scaler only on the training set. The class labels were removed from the input data to prevent accidental usage during training, helping to maintain data integrity. This ensured that only relevant features were used for unsupervised learning, optimizing memory and computational efficiency. The preprocessed data was then suitable for training models and evaluating their performance accurately on unseen data.

Model Selection

The first model used was the auto encoder. Auto encoders as shown in Figure 4 learn by encoding their input data to a lower or higher dimension in a hidden layer and try to reproduce the input by decoding back to the original input, through a process known as reconstruction. Since the hidden layers require the model to prioritize which properties are most important for reconstructing the output, they are often used for feature selection. The most significant features of the input data may be captured by the hidden layers with lower dimensions than the input layer. While properties such as robustness to missing inputs or to noise may be captured by hidden layers with higher dimensions than the input layer (Renström Timothy Holmsten Kth Skolan För Kemi & Och Hälsa, n.d.).

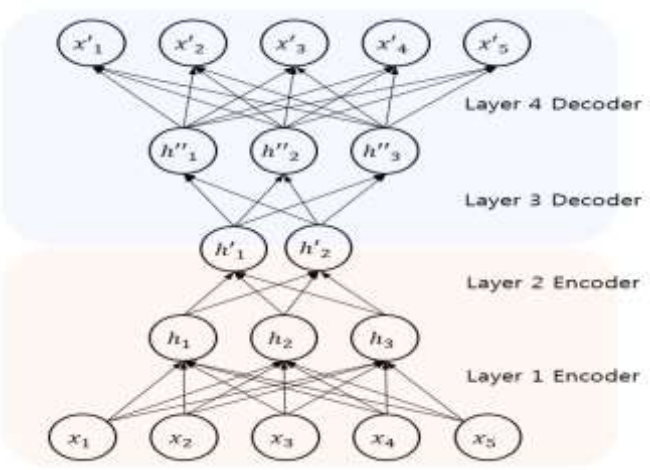

Figure 3: Autoencoder Architecture. (Pumsirirat & Yan, 2018)

The second model was the RNN. Recurrent Neural Networks are designed to handle sequential data, such as time series or natural language. RNNs can be thought of as an extension of feed forward neural networks that can process inputs of varying length and are capable of modeling temporal dependencies in the data. RNNs are ideally suited for speech recognition, language modeling and machine translation because they can capture long-term dependencies in the data. When processing long sequences, RNNs can be prone to vanishing gradients, which can affect the networks learning process due to the gradient shrinking exponentially if the weights are small. A type of RNN known as Long Short-Term Memory (LSTM) network was created to solve this problem. They work by introducing memory cells and gating mechanisms into standard RNN architecture. The memory cells can store information for long periods, and the gating mechanisms can selectively add or remove information from the memory cells. This allows LSTMs to remember important information from past inputs and selectively forget irrelevant information. Long sequences of transaction data can be analyzed by LSTMs to find subtle outliers that might point to fraudulent transaction in a credit card transaction dataset.

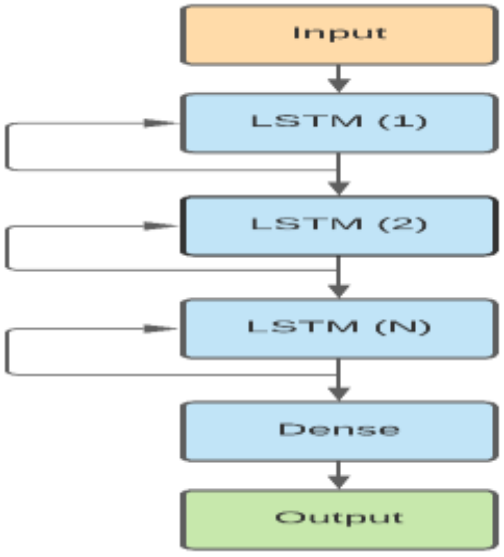

Figure 4: Recurrent Neural Network Architecture. (Maaliw et al., 2021)

The models were each trained with the Adam optimizer and Mean Squared Error (MSE) loss function to provide a robust framework for training. The Adam optimizer ensures fast and effective convergence during training due to its efficient computation and low memory requirements. It adjusts the learning rate for each parameter dynamically, based on the estimates of first and second moments of the gradients. This means that it helps the model's learning process, making sure the adjustments to the model are not too large or too small, but just right, based on both the immediate and past data it has seen. This leads to more efficient learning, often reaching better results faster than other optimization methods that don't adjust as intelligently. While MSE measures the average squared difference of the errors between the predicted values and the actual values. This implies that the MSE penalizes large errors more than smaller ones, leading to a more accurate reconstruction of the input data.

Training and Validation Process

Early Stopping was used to monitor the validation loss during training and stop the training process if the validation loss does not improve for five consecutive epochs, which helps to prevent over fitting by stopping the training when the model starts to overlearn on the training data. Model Checkpoint was also used to save the model that achieves the best performance, in terms of validation loss, to a specified file path, ensuring that the best version of the model is preserved even if the model's performance degrades in subsequent epochs.

The auto encoder was trained on 100 epochs with a batch size of 256. The training data is shuffled in each epoch to prevent the model from learning any potential order in the training data, which could lead to over fitting. The validation set is then used to evaluate the model's performance on unseen data after each epoch, providing insight into how well the model generalizes during training. The RNN was trained on 30 epochs with a batch size of 256. This time, the training data is not shuffled in each epoch because maintaining the chronological order of the dataset was crucial for the RNN to learn temporal patterns.

From the graph in Figure 5, the training loss decreases sharply at the beginning and then continues to decrease at a slower rate as the number of epochs increases. This is expected as the model begins to learn from the training data. The validation loss also decreases, but after a certain point, it plateaus and does not significantly improve with more epochs. This is indicative of the model starting to converge and suggests that additional training beyond this point may not result in substantial improvements on the validation set. Since the validation loss did not increase, it can be concluded that early stopping successfully prevented the auto encoder from over fitting during training.

From the graph in Figure 6, the RNN may not be generalizing well to the validation data from the start. The validation loss plateaus quickly and is not decreasing, suggesting that the improvements the model is making during training are not translating to better performance on unseen data.

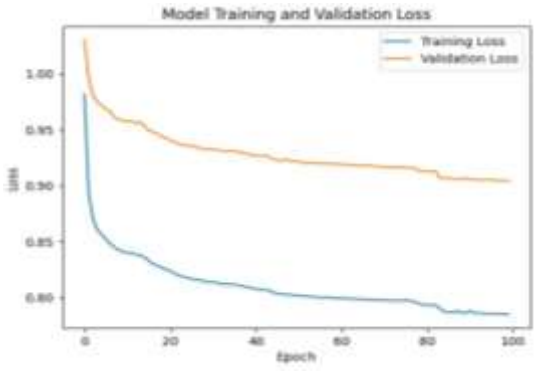

Figure 5: Auto encoder Loss Graph

Figure 6: RNN Loss Graph

Ensemble Learning Approach

Ensemble learning is a process whereby various learning models or algorithms are combined to obtain a single yet more powerful and improved model than the individual ones (Mohammed & Kora, 2023). The diversity of each of the single models makes ensemble learning capable of reducing the risk of over fitting. The idea behind ensemble learning is the integration of the outputs from each baseline model to form one single output in a new model. One of the ways of doing this is using a strategy called Voting Method which consists of three types:

Max Voting: Here, predictions are obtained from each model and the prediction that occurs the most will be the output.

Average Voting: Like max voting, however, this time the arithmetic mean of the all classifiers' predictions is used to determine the final prediction.

Weighted Average Voting: Like Average voting, but different weights are given to each model, indicating their importance in the prediction.

After successfully training the auto encoder and RNN, predictions from both models were generated, and then by computing their MSE, the reconstruction error of each model was measured. To determine the ensemble strategy that will give the best result, all three voting methods were performed for comparison.

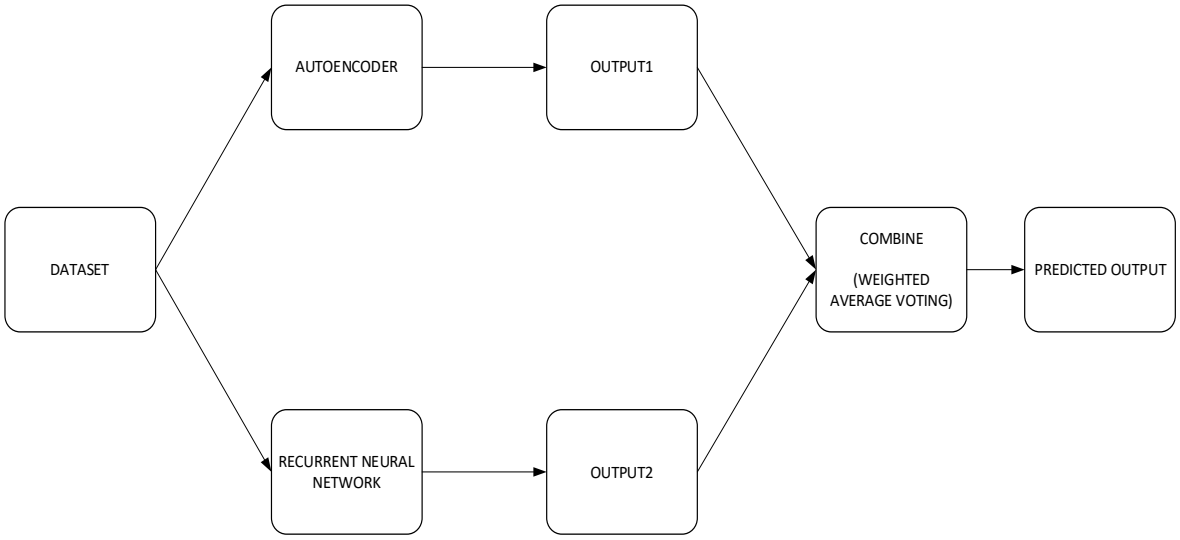

Figure 7: Ensemble Model Architecture

Evaluation Metrics

After successfully training the models, it was necessary to evaluate the performance of the individual models prior to the ensemble and evaluate the performance of the ensemble model to determine how well it performs over the individual models. Accuracy metric is the most common performance metric for model evaluation; it measures the proportion of total correct predictions (true positives and true negatives) out of the total number of predictions present. However, because the dataset used in this work was unbalanced, accuracy was not considered as one of the performance metrics. This is because accuracy metric in the context of an unbalanced dataset can be misleading; a model could trivially predict the majority class for all instances and still achieve high accuracy, while failing to correctly identify the minority class. Fraudulent transactions, which are the minority class, were of greater interest; therefore it was more informative to use other metrics that could provide a more nuanced view of the model's performance, such as: precision, recall, F1-score, and Area under the Receiver Operating Characteristic Curve (AUC-ROC).

Precision: This measures the proportion of true positives among all samples predicted as positive. A high precision indicates that when a model predicts the positive class, it is correct a high proportion of the time. This is especially important when the cost of false positives is high.

$$\text{Precision} = \frac{TP}{(\text{TP}+\text{FP})} \quad (1)$$

Recall: This can also be referred to as sensitivity. It is the ratio of true positives to the sum of true positives and false negative. It's crucial in scenarios where missing the positive samples (false negatives) is costly. A high recall indicates that the model is effective in identifying the minority class.

$$\text{Recall} = \frac{TP}{(\text{TP}+\text{FN})} \quad (2)$$

F1 Score: This is the harmonic mean of the precision and recall scores obtained for the positive class.

$$\text{F1 Score} = \frac{2 * (\text{Precison} * \text{Recall})}{(\text{Precion} + \text{Recall})} \quad (3)$$

Area Under the Curve (AUC): This is a single number summarizing the information of the ROC curve. It quantifies the overall ability of the model to distinguish between classes regardless of the class distribution. AUC ranges from 0 to 1. An AUC of 0.5 suggests no discriminative ability (equivalent to random guessing). The higher the AUC, the better the model is at distinguishing between positive and negative classes across all possible thresholds, but it does not specify at which threshold the model performs best.

RESULTS AND DISCUSSION

The results from the study highlight the varying strengths of the individual models and the improvements achieved through ensemble techniques. The Auto encoder performed well with a high recall of 0.92 and an AUC of 0.96, indicating strong capabilities in detecting fraudulent transactions but suffered from low precision (0.052), leading to a high rate of false positives. The RNN demonstrated moderate performance with a precision of 0.38, recall of 0.24, F1-score of 0.30, and an AUC of 0.82, revealing balanced but generally lower performance metrics. Among the ensemble methods tested, Weighted Average Voting provided the best results, achieving a precision of 0.24, recall of 0.51, F1-score of 0.33, and an AUC of 0.95, which balanced the strengths of the individual models. Comparatively, the ensemble method improved over Pumsirirat's Auto encoder in precision and F1-score, matched the AUC of the RBM model, and performed competitively with Rezapour's models, though direct comparisons were limited by differences in evaluation metrics. The discussion highlights that the ensemble model effectively balances precision and recall while improving overall performance, though slight reductions in AUC compared to the Auto encoder suggest a trade-off due to errors from the RNN model. This ensemble approach shows promise by enhancing the strengths of individual models and providing a balanced performance in fraud detection tasks.

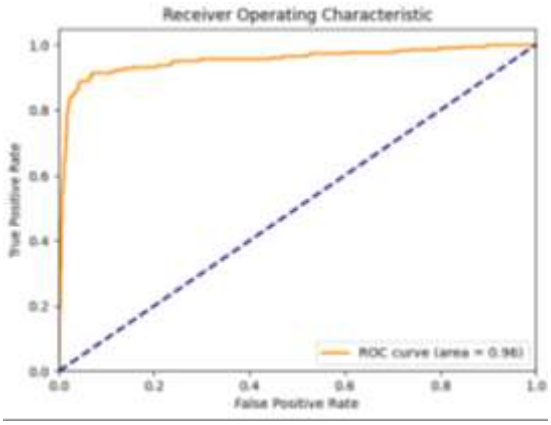

Figure 8: AUC of Auto encoder

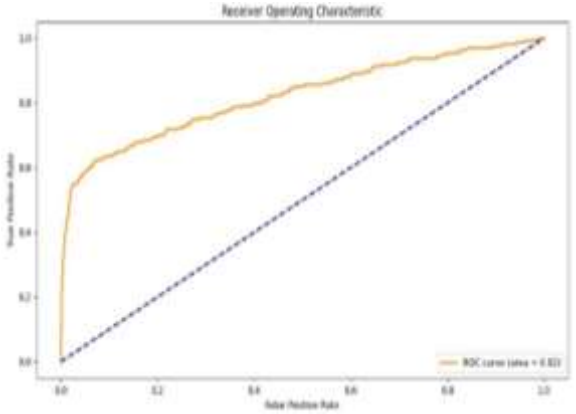

Figure 9: AUC of the RNN

Table 1: Classification Report of the Auto encoder and RNN

Model	Precision	Recall	F1-Score
Auto encoder	0.052	0.92	0.097
RNN	0.38	0.24	0.30

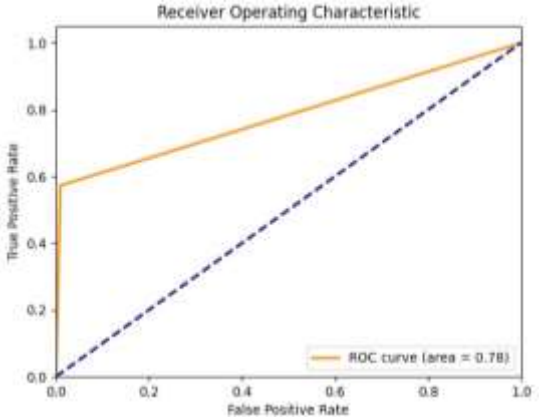

Figure 10: AUC for Max Voting

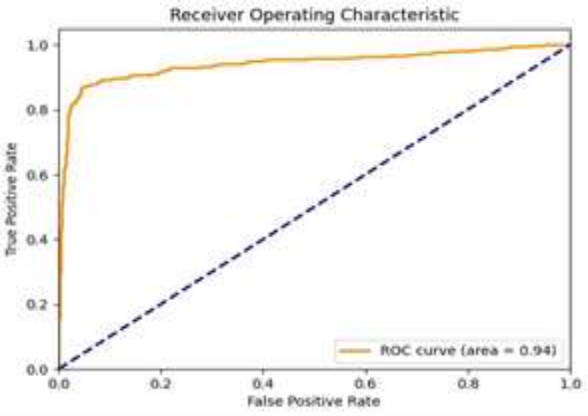

Figure 11: AUC for Average Voting

Table 2: Comparison with Existing Works

Author	Model	Dataset	Precision	Recall	F1-Score	AUC
A.Pumsirirat	Autoencoder	European Dataset	0.047	0.837	0.089	0.96
	RBM	European Dataset	-	-	-	0.95
M.Rezapour	One-class SVM	European Dataset	0.69	0.86	0.76	-
	Autoencoder	European Dataset	0.91	0.87	0.89	-
	Robust Mahanabolis	European Dataset	0.68	0.50	0.58	-
Proposed Approach	Autoencoder	European Dataset	0.052	0.92	0.097	0.96
	RNN	European Dataset	0.38	0.24	0.30	0.82
	Ensemble	European Dataset	0.24	0.51	0.33	0.95

Discussion

The results obtained in this study showcases the strengths and weaknesses of the individual models in different areas, and the improvements made by the ensemble model. The Autoencoder excels in recall and AUC, it is highly effective in identifying most of the fraudulent transactions present in the dataset and it has a strong capability to differentiate between fraudulent and non-fraudulent transactions across different thresholds. However, it suffers from low precision, making it prone to generate a high number of false positives. While the RNN presents a moderate performance across the board, with its primary challenge being a low recall. The Ensemble model enhances the strengths and overall performance of the individual models, with a promising balance in precision and recall. It significantly improved the AUC of the RNN model, but slightly reduced that of the Autoencoder, suggesting that possible errors from the RNN could have negatively impacted the Autoencoder's performance when combined.

Most research works on unsupervised learning did not evaluate their models using all four of the evaluation metrics that were used in this research, limiting a direct comparison of this model's performance with theirs. However, from the results in Table 2, this Autoencoder shows an improvement over Pumsirirat's (Pumsirirat & Yan, 2018) Autoencoder when giving the same threshold to classify transactions. The

Ensemble model matches the author's RBM's AUC score and beats his Autoencoder in terms of precision and F1 score. The models from Rezapour's (Rezapour, 2019) work have a good balance in terms of precision and recall; however, they did not measure their AUC score, which is important in determining the model's overall ability to differentiate between fraudulent and normal transactions across various thresholds. It can be argued that the author's models were overfitting because they were trained and tested on the same dataset, without splitting.

CONCLUSION

Unsupervised learning is very important in the field of credit card fraud detection due to its ability to update itself without relying on labeled data, making it able to detect new types of fraud as soon as possible. From the results obtained in this study, the ensemble model showed potential to be effective in improving the performance of unsupervised learning models in this field, as it showed better results compared to research work that carried out the same task, on the same dataset, with the same model. To the best of our knowledge, the unsupervised RNN model used in this study has not been used in any literature for credit card fraud detection before, therefore this work made another contribution to the field by introducing a different technique and improved the performance of that technique. However, there is still a need for improvement in balancing the precision and recall of our Ensemble model, to reduce false positives while maintaining high detection rates.

Further work can be undertaken to attempt to improve the balance of this model's precision and recall, as well as the overall performance of the model. Such might include feature engineering to add new and relevant features to the dataset, more fine tuning of the model's hyper parameters, adding or removing layers from the individual models, etc. Future researchers can also implement this model in real-time and add a user feedback system to prevent false positives and negatives.

REFERENCES

Al-Faiz, M., Ibrahim, A., & Hadi, S. (2019). The effect of Z-Score standardization (normalization) on binary input due to the speed of learning in back-propagation neural networks. *Iraqi Journal of Information & Communications Technology, 1*, 42–48.

Bodepudi, H. (2021). Credit card fraud detection using unsupervised machine learning algorithms. *International Journal of Computer Trends and Technology, 69*(8), 1–3.

Boucher, É. (n.d.). *Outlier detection methods applied to financial fraud*.

Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (n.d.). *Combining unsupervised and supervised learning in credit card fraud detection*.

Confusion matrix: How to use it & interpret results [Examples]. (n.d.). Retrieved January 21, 2024, from https://www.v7labs.com/blog/confusion-matrix-guide

Credit card fraud detection | Kaggle. (n.d.). Retrieved April 19, 2023, from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Hassan, H., Ahmad, M. A., & Mustapha, R. (2024). An enhanced feature engineering technique for credit card fraud detection. *FUDMA Journal of Sciences, 8*(4), 8–16.

Islam, M. A., Uddin, M. A., Aryal, S., & Stea, G. (2023). An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes. *Journal of Information Security and Applications, 78*.

Maaliw, R. R., Mabunga, Z. P., & Villa, F. T. (2021). Time-series forecasting of COVID-19 cases using stacked long short-term memory networks. In *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2021* (pp. 435–441).

Mohammed, A., & Kora, R. (2023a). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences, 35*(2), 757–774.

Mohammed, A., & Kora, R. (2023b). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences, 35*(2), 757–774.

Niu, X., Wang, L., & Yang, X. (n.d.). *A comparison study of credit card fraud detection: Supervised versus unsupervised*. www.aaai.org

Oghenekaro, L., & Ugwu, C. (2016). A novel machine learning approach to credit card fraud detection. *International Journal of Computer Applications, 140*, 45–50.

Pumsirirat, A., & Yan, L. (2018). Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine. *International Journal of Advanced Computer Science and Applications (IJACSA), 9*(1). www.ijacsa.thesai.org

Renström, T. H. (n.d.). *Fraud detection on unlabeled data with unsupervised machine learning*. KTH School of Chemistry, Biotechnology and Health.

Rezapour, M. (2019). Anomaly detection using unsupervised methods: Credit card fraud case study. *International Journal of Advanced Computer Science and Applications (IJACSA), 10*(11). www.ijacsa.thesai.org

Saraf, S., & Phakatkar, A. (n.d.). Detection of credit card fraud using a hybrid ensemble model. *International Journal of Advanced Computer Science and Applications (IJACSA), 13*(9). www.ijacsa.thesai.org

Understanding AUC - ROC curve | by Sarang Narkhede | Towards Data Science. (n.d.). Retrieved January 22, 2024, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

West, J., Bhattacharya, M., & Islam, R. (n.d.). *Intelligent financial fraud detection practices: An investigation*.

Zhang, A., Lipton, Z. C., Li, M. U., & Smola, A. J. (n.d.). *Dive into deep learning*.