# Autoencoders with Reconstruction Error and Dimensionality Reduction for Credit Card Fraud Detection

Najmi Rosley[1], Gee-Kok Tong[1(✉)], Keng-Hoong Ng[1], Suraya Nurain Kalid[1], and Kok-Chin Khor[2]

[1] Multimedia University, Cyberjaya, Malaysia
gktong@mmu.edu.my
[2] Universiti Tunku Abdul Rahman, Petaling Jaya, Malaysia

**Abstract.** The increase in credit card transactions has inevitably caused an increase in credit card fraud. A total of 157,688 fraud cases occurred in 2018 worldwide, causing a total loss of $24.26 billion. This paper proposes using two types of autoencoder models to detect credit card fraud. The first type uses reconstruction error to detect anomalies in the data. The model detects fraud by defining a threshold in the reconstruction error to flag the transactions as legitimate or fraud. The second type performs dimensionality reduction to encode the data and removes noises. The encoded data were then used to train three other models: K-nearest neighbours (KNN), logistic regression (LR), and support vector machine (SVM). We then applied these models to a European bank's imbalanced credit card data set. A comparison was made between the two autoencoder types and three baseline models: KNN, LR and SVM. The results showed that both autoencoders gave a good and comparable performance in detecting credit card frauds.

**Keywords:** Autoencoder · Credit Card Fraud Detection · Reconstruction Error · Dimensionality Reduction

## 1 Introduction

During the pandemic, credit card transactions have increased dramatically due to cashless transactions between cardholders and e-commerce companies. Although they have benefited from these cashless transactions, new problems arise. One of the problems is the increasing fraudulent transactions. There was a total of 157,688 fraud cases recorded worldwide in 2018. The damages caused by such frauds totalled $24.26 billion [1].

This increasing number of credit card frauds has motivated the need to develop an effective fraud detection system. The system must be able to detect fraudulent credit transactions accurately and reliably with high true positive and minimal false negative rates. Much research has been conducted about building a machine learning model that can detect fraud transactions [2–4]. Nevertheless, the research works suffered from the problems of poor classification performance, specifically for the minority class. Most


S.-C. Haw and K. Sonai Muthu (Eds.): CITIC 2022, 10, pp. 503–512, 2022.
https://doi.org/10.2991/978-94-6463-094-7_40

machine learning algorithms used for classifications were designed around the assumption of an equal number of examples for each data set class. Usually, credit card transaction data are imbalanced and have more legitimate transactions than fraudulent ones. As a result, the available public credit card data sets are mostly highly imbalanced. Such imbalance has become a problem because the performance of classification algorithms is evaluated using accuracy, but in the case of the imbalanced data set, the minority class contributes little to the overall accuracy of the model [5]. The imbalance can cause the trained model to bias toward the majority case.

This research aims to detect credit card frauds in an imbalanced data set from a European bank [6] with autoencoders. Firstly, we created a supervised machine learning model—an autoencoder via reconstruction error. Secondly, we conducted dimensionality reduction using autoencoders to improve the performance of the supervised machine learning models, i.e., KNN, LR, and SVM. The research scope shall cover data cleaning, feature selection, hyperparameter tuning, autoencoder model building, and model evaluation.

## 2 Related Works

### 2.1 Autoencoder

Autoencoder is a neural network that learns to recreate the input as an output. A simple autoencoder consists of the input, the bottleneck and the output layers [7]. The model extracts the main features and distribution of the input data by decoding and encoding the input data. The main goal of the autoencoder is to approximate the distribution of the input value as accurately as possible (Fig. 1).

The encoder reduces the dimensionality of the input data to a lower dimensionality. The bottleneck is one of the hidden layers of the autoencoder in which the compressed data representation is stored. Simple autoencoders only have one hidden layer, but advanced autoencoders may have more hidden layers. The decoder decodes or increases

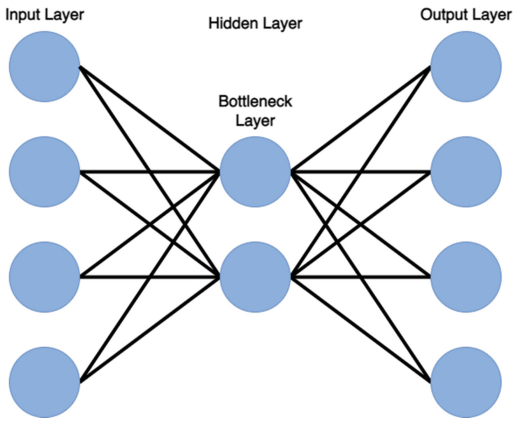


**Fig. 1.** The Autoencoder Diagram. Adopted from [7].

the data dimensionality stored in the bottleneck and tries to reconstruct the input. The reconstruction error will be used to evaluate the performance of the autoencoder. The reconstruction error measures how well the input is reconstructed by measuring the input and output difference. The bottleneck layer holds the latent representation of the input data, and the lower dimensionality representation of the input data can be obtained by extracting the data from the bottleneck layer. For imbalanced data classification using autoencoders, most researchers use two common implementations. The first is by using sampling techniques such as the work done by [8], in which a cost-sensitive oversample algorithm is used in conjunction with autoencoders to oversample the minority class and remove noise from multiple imbalanced medical datasets. The other implementation is treating the problem as an anomaly detection problem whereby the model only learns the representation of the majority class. Such as the research done by [9] to classify depression in an imbalanced dataset.

### 2.2 Thresholding Based on Reconstruction Error of Autoencoders

Much research has been conducted in which the reconstruction error value of the autoencoder can be used for anomaly detection. The model is trained only using the normal data. It is to ensure that the reconstruction error for reconstructing normal data is low, and when reconstructing anomalous data, the reconstruction error shall be high. Thus, a reconstruction error threshold can be selected to identify anomalies in data. Research by [10] used a reconstruction error threshold to detect anomalies in network traffic that indicate cyberattacks. Another research by [11] used an autoencoder to detect anomalies in vibration signals made by rotating machines. Their model was trained by using only normal data in which the machines are in a healthy state and working properly. The research helps maintain machines' health by detecting when the machines malfunction and need repairs.

### 2.3 Dimensionality Reduction with Autoencoders

Autoencoders also can be used as a dimensionality reduction model. It is due to the nature of autoencoders during the encoding phase, in which autoencoders can identify the important features and remove noises in data. The lower dimensionality representation of data can be extracted from the bottleneck layer and used to train other machine learning models. Hence, autoencoders can be used to improve the performance of other machine learning models. Research by [3] made a fraud detection model for credit card transactions using an autoencoder and three classification algorithms: Multilayer-Perceptron, K-Nearest Neighbours, and Logistic Regression. The results showed that ensembling autoencoders and k-Nearest Neighbour gave the best performance with an accuracy of 0.9995 and an F1-score of 0.8182. Another research by [12] built an ensemble autoencoder model to detect anomalies in building energy data. The model used an autoencoder to reduce the data dimension and remove noises for better detection. The model was trained using unsupervised learning, and the Shapiro Wilk test was carried out to evaluate the normality of the reconstructed residuals and evaluate the model's performance. The ensemble model was able to pass the normality test, which indicates that the models could reconstruct the data accurately.

### 2.4 Feature Selection and Hyperparameter Tuning of Autoencoders

Feature selection refers to selecting the data set features because a large number of features will slow down the training process and may not necessarily improve an autoencoder's performance. Besides this, much research has been carried out to improve the effectiveness and accuracy of autoencoders by using hyperparameter tuning. Hyperparameter tuning ensures an autoencoder is adequately tuned and set to give its best performance. A study by [13] has demonstrated that the hyperparameter tuning of autoencoders can be automated using H20 AI to avoid manual tuning, which is very time-consuming. Besides using AI, there are Python libraries that help achieve such automation. One such library is Keras Tuner, used in this research. Research by [14] built multiple machine learning models using different feature analysis techniques to determine which one gave the best accuracy. The research showed that feature extraction methods gave neglectable decreases in the model's accuracy compared to the control experiment, where a classifier using all 325 potential features was trained and tested. Contrarily, using the wrapper methods with Ranked Forward Search showed a considerable improvement in the model's accuracy.

### 2.5 Related Works Summary

Other researchers have given insight into possible solutions to the credit card fraud detection problem. The results obtained by other researchers have proven that autoencoders can overcome the imbalanced data problem with satisfactory results in other applications. In this research, both methods (sampling method and anomaly detection method) were tested to measure their effectiveness in overcoming this problem.

## 3 Methodology

In the following sections, we shall explain our research framework based on Fig. 2.

### 3.1 Data Collection

This research uses the European credit card data set. The European data set was obtained from Kaggle, a popular platform for finding data sets on various topics [6]. The European credit card data set was formed by collecting two-day transaction data in September 2013. The data set contains 284,808 transactions. The data set is imbalanced as it only has 492 transactions labelled as frauds which only make up 0.172% of the total transactions. Most features available in this data set had been transformed using Principal Component Analysis (PCA) to hide the private information. However, some features are not hidden. Firstly, the "time" feature denotes the number of seconds a transaction has been carried out since the first transaction. Secondly, the "amount" feature details the amount of money used in the transaction. Lastly, the "class" label indicates the legitimacy of the transactions. 0 denotes that the transaction is legitimate, and 1 denotes that the transaction is fraudulent.

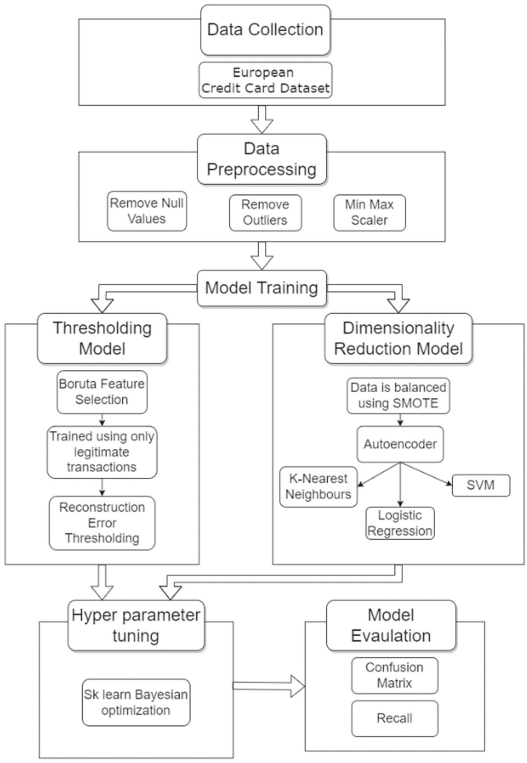


**Fig. 2.** Our Research Framework for Credit Card Fraud Detection

### 3.2 Data Pre-processing

Data pre-processing was performed to ensure that the data was clean, consistent, and meaningful for training the autoencoders. The outliers were removed from the legitimate data to avoid overfitting in the model training phase. The outliers were identified by calculating the Z-score of the data. Only data with a z-score of less than three were kept. Then, the data were normalised using min-max scaling to ensure that the data had the same scale. Normalisation is necessary because the model may bias toward features with larger values.

### 3.3 Feature Selection with Boruta

Boruta was used in this research as a means for feature selection. The features were fitted using a random forest classification in **Boruta**. The feature selection process was repeated ten times to ensure that the most accurate combination of features was obtained. Only features with an importance score of 0.5 or higher were kept and used to train the autoencoder for each iteration.

**Table 1.** Hyperparameter tuning range and the best values for the autoencoders.

| Hyperparameters | Range | Best Value |
|---|---|---|
| Number of neurons in hidden layer | 10–20 | 10 |
| Epochs | 10–30 | 30 |
| Batch Size | 32–128 | 32 |

### 3.4 Hyperparameter Tuning

The hyperparameter tuning of autoencoders was performed using the Bayesian search function in Sklearn. The hyperparameters tuned were the number of nodes in the hidden layers, the activation function, the EPOCH, and the batch size. The hyperparameter tuning range and the best values are shown in Table 1.

### 3.5 Model Training

Two models were trained in this research using two different datasets: (i) the reconstruction error thresholding model and (ii) the dimensionality reduction model. The thresholding model was trained using only legitimate transaction data, while the dimensionality reduction model was trained using the entire dataset. Thus, their hyperparameters were tuned separately.

#### 3.5.1 Reconstruction Error

After data pre-processing and feature selection with Boruta, the data were split into train and test sets (80–20 split). Only the legitimate transactions from the training data were used to train an autoencoder; this ensures a low reconstruction error for legitimate transactions. After that, the autoencoder was used to predict the test set, and from there, we can plot out the reconstruction error for the fraud and legitimate transactions. The classification threshold is selected using the precision-recall curve to maximise the precision and recall scores of the model.

#### 3.5.2 Dimensionality Reduction

This autoencoder type reduces the data set dimension and yields encoded data. The autoencoder was used in conjunction with these three machine learning models: k-Nearest Neighbours (KNN), Support Vector Machine (SVM), and Logistic Regression (LR). These three models were trained using the encoded data. SVM, KNN and LR were chosen as they are the popular machine learning methods used by other researchers for credit card fraud detection [15–17].

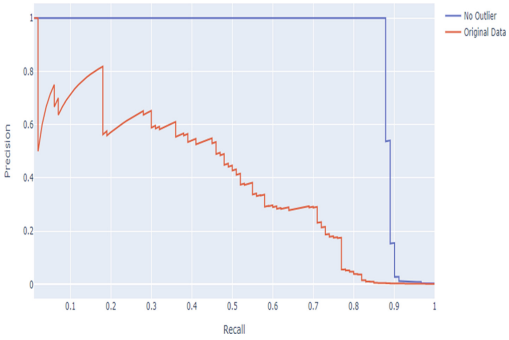


**Fig. 3.** The Precision-Recall Curve

### 3.6 Model Evaluation

All the autoencoders were compared with the performance of the baseline models: KNN, SVM, and LR, to determine if the autoencoders could help improve credit card fraud detection. The baseline models were trained using the data set reduced by Boruta but unencoded.

All models were evaluated using measures resulting from the confusion matrix: accuracy, precision, recall and F1. Of these four measures, recall is used to evaluate the performance of the models, particularly in detecting minority credit card fraud in the data set.

## 4 Results and Discussion

Figure 3 shows the precision-recall curve that selects the threshold from the reconstruction errors of two data sets. The blue line represents the data set whereby the outliers in the legitimate transaction data had been removed, while the orange line represents the original credit card data set with outliers. The data set with the outliers removed performed better as it can produce high recall and precision values. For this research, a reconstruction error threshold of 0.074 was used to classify between the fraud and legitimate transactions.

Table 2 compares the model performance using accuracy, precision, recall, and F1 score. All models showed no significant difference in accuracy, precision, and F1 score. Thus, we relied on the recall measure to differentiate the performance of the models. The autoencoders that used reconstruction error performed relatively well compared to others, with the highest recall of 0.879121. However, its performance was not much different from the autoencoders that performed dimensionality reduction and were used in conjunction with KNN, LR and SVM. Baseline models generally showed a below-average performance, except for KNN.

**Table 2.** Performance comparison between the autoencoders and the baseline models

| Models | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Autoencoder (Reconstruction Error) | **0.999775** | **1.0** | **0.879121** | **0.935673** |
| Autoencoder (Dimensionality Reduction)+KNN | 0.999673 | 1.0 | 0.850467 | 0.919191 |
| Autoencoder (Dimensionality Reduction)+LR | 0.999693 | 1.0 | 0.827586 | 0.905660 |
| Autoencoder (Dimensionality Reduction)+SVM | 0.999673 | 1.0 | 0.816091 | 0.898734 |
| Baseline KNN | 0.999632 | 1.0 | 0.831776 | 0.908163 |
| Baseline LR | 0.999469 | 1.0 | 0.757009 | 0.861702 |
| Baseline SVM | 0.999550 | 1.0 | 0.794393 | 0.885417 |

## 5 Conclusion

In conclusion, the autoencoders that (i) used reconstruction error and (ii) performed dimensionality reduction gave a good performance in detecting credit card fraud. It is due to the autoencoders' feature extraction capability to learn the data's inner representations.

We see the possibility of employing autoencoders on imbalanced data sets and obtaining good detection performance. In future, we shall further evaluate the performance of autoencoders using more imbalanced credit card data sets.

**Acknowledgements.** The Ministry of Higher Education Malaysia supported this work under Grant FRGS/1/2019/SS01/MMU/03/11.

## References

1. Credit Card Fraud Statistics. Shift Credit Card Processing. (2021, September 29). Retrieved April 11, 2022, from https://shiftprocessing.com/credit-card-fraud-statistics/
2. Shivamb. (2019, January 18). Semi supervised classification using autoencoders. Kaggle. Retrieved October 13, 2021, from https://www.kaggle.com/shivamb/semi-supervised-classification-using-autoencoders
3. Misra, S., Thakur, S., Ghosh, M., & Saha, S. K. (2020). An autoencoder based model for detecting fraudulent credit card transaction. Procedia Computer Science, 167, 254–262. https://doi.org/10.1016/j.procs.2020.03.219
4. Al-Shabi, M. A. (2019). Credit card fraud detection using Autoencoder model in unbalanced datasets. Journal of Advances in Mathematics and Computer Science, 1–16. https://doi.org/10.9734/jamcs/2019/v33i530192
5. Gosain, A., & Sardana, S. (2017). Handling class imbalanced problem using oversampling techniques: A Review. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). https://doi.org/10.1109/icacci.2017.8125820

6. ULB, M. L. G.-. (2018, March 23). Credit Card Fraud Detection. Kaggle. Retrieved April 11, 2022, from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
7. Zhang, G., Liu, Y., & Jin, X. (2019). A survey of autoencoder-based Recommender Systems. Frontiers of Computer Science, 14(2), 430–450. https://doi.org/10.1007/s11704-018-8052-6
8. Zhang, C., Gao, W., Song, J., & Jiang, J. (2016). An imbalancedd data classification algorithm of improved autoencoder Neural Network. 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI). https://doi.org/10.1109/icaci.2016.7449810
9. Gerych, W., Agu, E., & Rundensteiner, E. (2019). Classifying depression in imbalancedd datasets using an autoencoder- based anomaly detection approach. 2019 IEEE 13th International Conference on Semantic Computing (ICSC). https://doi.org/10.1109/icosc.2019.8665535
10. Salahuddin, M. A., Faizul Bari, M., Alameddine, H. A., Pourahmadi, V., & Boutaba, R. (2020). Time-based anomaly detection using autoencoder. 2020 16th International Conference on Network and Service Management (CNSM). https://doi.org/10.23919/cnsm50824.2020.9269112
11. Ahmad, S., Styp-Rekowski, K., Nedelkoski, S., & Kao, O. (2020). Autoencoder-based condition monitoring and anomaly detection method for rotating machines. 2020 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/bigdata50022.2020.9378015
12. Fan, C., Xiao, F., Zhao, Y., & Wang, J. (2018). Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. Applied Energy, 211, 1123–1135. https://doi.org/10.1016/j.apenergy.2017.12.005
13. Ordway-West, E., Parveen, P., & Henslee, A. (2018). AUTOENCODER evaluation and hyper-parameter tuning in an unsupervised setting. 2018 IEEE International Congress on Big Data (BigData Congress). https://doi.org/10.1109/bigdatacongress.2018.00034
14. Shardlow, M. (2016). An Analysis of Feature Selection Techniques.
15. Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis. *2017 International Conference on Computing Networking and Informatics (ICCNI)*. https://doi.org/10.1109/iccni.2017.8123782
16. Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019). Real-time credit card fraud detection using machine learning. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). https://doi.org/10.1109/confluence.2019.8776942
17. Preeti, S., A. (2019). Analysis of various credit card fraud detection techniques. International Journal of Computer Sciences and Engineering, 7(6), 1212–1216. https://doi.org/10.26438/ijcse/v7i6.12121216