Detection of Credit Card Fraud in a Brazilian database using Autoencoder Neural Network ⋆

Frederico Luis de Azevedo, Karin Satie Komati, Hilário Seibel Júnior

Programa de Pós-graduação em Computação Aplicada (PPComp)
Instituto Federal do Espírito Santo (IFES), Campus Serra
(e-mail: fredazevedo89@gmail.com, {kkomati,hsjunior}@ifes.edu.br)

Abstract: The increasing number of credit-card transactions made over the internet in recent years has lead to a rise in the same proportion in the amount of fraud. Due to the large volume of web-based transactions that should be carried out daily, it is necessary to have a robust system to predict such crime to reduce loss and increase the confidence of banks and issuers. Deep Learning techniques emerge as a way to automate this process, training classifiers with data from past transactions to try to predict future frauds. In this paper, we build an Autoencoder model and perform a threshold tuning to predict fraudulent transactions. A proprietary Brazilian credit-card transaction database was used for training and performance evaluation of the model, containing almost 40 million transactions and challenging frauds, which were not previously detected by the organization's current fraud detection systems. The results of the experiments presented satisfactory results in a real-world dataset. The Autoencoder metrics show a good performance in fraud classification, reaching a positive Matthews Correlation Coefficient value and an AUC of 0.81, which are not affected by the database imbalance.

Keywords: credit-card fraud detection; deep learning; autoencoders; Matthews Correlation Coefficient.

1. INTRODUCTION

The growth in the number of online credit-card transactions, caused mainly by the popularity of e-commerce in the past decade, has made cases of fraud even more common. A report published by Knieff (2016) states that more than 45% of survey respondents residing in the United States, Mexico and Brazil have suffered some form of card fraud in the past 5 years. The study also affirms that these countries are favorable environments for such types of attacks as e-commerce companies do not have strong controls for preventing fraud. Implementing effective fraud-detection solutions is of extreme importance for all organizations issuing credit cards or managing online transactions, in order to reduce losses and, at the same time, to improve customers' confidence (Fiore et al., 2019).

The appropriate moment for an information system to detect fraud in a credit-card transaction is during the authorization phase. The credit-card authorization process involves at least five parties: the cardholder, the merchant, the acquirer, the credit-card network, and the issuing bank (each of them is responsible for an authorization step). Figure 1 illustrates the authorization flow and the relationship between the parties involved in the process. The credit card issuer, often banks or fintech, is responsible for giving the final acceptance of purchase during the last step in the authorization flow.

It might be more reliable when it is possible to incorporate fraud detection into the issuer information system by adapting it to the needs of the organization and to the customers purchasing profile. Techniques involving machine learning prove to be effective to meet this challenge (Zhang et al., 2019). The approach that is gaining importance is known as Deep Learning, a technique that was initially used in automatic speech recognition, image recognition and more recently in the detection of fraud in financial transactions according to Roy et al. (2018). A paper by Breslow et al. (2017) mentions that Deep Learning need large amounts of data and well-tuned models, predicting that this technique will likely begin to be deployed on a large scale by banks in the coming years to combat money laundering, fraud and other financial crimes.

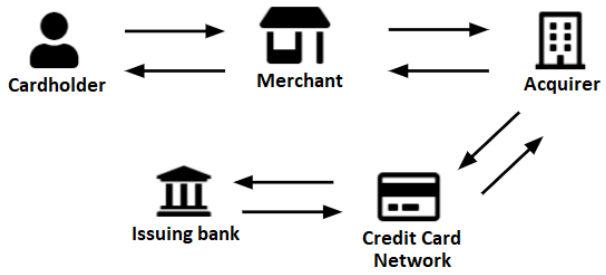

Figure 1. Credit-card authorization flow.

An Autoencoder is a type of artificial neural network used to learn data in an unsupervised manner and suitable for credit-card fraud detection. Although the research of Pumsirirat and Yan (2018) also uses an Autoencoder model to detect credit-card fraud, this study differs by proposing the definition of a threshold to make the classification based

⋆ Agradecemos ao Propós (Programa Institucional de Apoio à Pós-graduação Stricto Sensu) do IFES pela apoio financeiro.

on the model result. The threshold can also be configured at any time by the credit-card issuer itself to be more restrictive or not regarding the classification of frauds. There is no extra processing cost by changing the threshold value since the model does not need to be re-trained. This offers more flexibility to the issuer in relation to the level of experience and customer impact it wants to obtain.

The methodology in this investigation of Autoencoder' threshold tuning is quite similar to that presented by Al-Shabi (2019). But in the paper of Al-Shabi (2019), the dataset was the Machine Learning Group of ULB (Université Libre de Bruxelles), which contains credit-card transactions from two days in Europe, composed by 284,315 legitimate transactions and 492 fraudulent transactions. But what if the database is different, with Brazilian and non-European data, would the results be similar?

Therefore, the objective of this paper is to investigate threshold tuning in the Autoencoder model in a database with credit-card transactions provided by a Brazilian fintech. This threshold tuning of the Autoencoder model in a proprietary Brazilian real-world dataset is a contribution of this study. It will be possible to analyze whether the variation of the Autoencoder threshold behaves in the same way even in different databases.

Section 2 of this article presents theme-related papers. Section 3 describes the dataset and performance metrics. Section 4 presents the Autoencoder model built and threshold tuning. Section 5 discusses the experiments and their results. Finally, Section 6 closes with the conclusion.

2. RELATED WORK

The work in Awoyemi et al. (2017) presents a comparative study among three machine learning techniques on an unbalanced database for fraud detection: Naive Bayes, k-Nearest Neighbor (kNN) and Logistic Regression. Their database is from European cardholders and contains 284,807 card transactions with 492 transactions labeled as frauds. 70% of the dataset is used for training while 30% is set for validating and testing. Each one of the techniques are applied on the database in its original, un-sampled form and also in other two datasets sampled in a hybrid approach where the positive class is oversampled and the negative class is under-sampled. The study concludes that the kNN had better performance for all metrics, achieving an Accuracy of 0,97 on the sampled and 0,96 on the un-sampled dataset.

The related papers described below uses Autoencoder for the classification of fraudulent and non-fraudulent transactions.

The research in Pumsirirat and Yan (2018) used three different credit-card transaction databases (German, Australian, and European) to validate two methods of Deep Learning: Autoencoders and Restricted Boltzmann Machines (RBMs). Both methods are unsupervised learning models for anomaly detection. Each model was individually evaluated with each database. The metrics used by the authors were the Mean Squared Error (MSE), the Root-Mean-Squared Error (RMSE), and Receiver Operating Characteristic Curve (ROC curve). The authors concluded that the Autoencoder model has a better performance than the Restricted Boltzmann Machine, especially in larger databases, such as the European. The German and Australian databases didn't achieve good results because they are very small and therefore not appropriate for Deep Learning models.

The purpose of the Rezapour (2019) paper was to study the behavior of applying three unsupervised learning methods for detecting credit-card fraud. The methods applied were One-Class SVM, Autoencoder, and Multivariate Outlier Detection, this one using the Mahalanobis Distance as a measurement for classification. The author indicates that the database is unbalanced since fraud corresponds to 0.17% of the total records. To overcome this problem, a random undersampling was applied to balance the database. The study concludes that the Autoencoder model was the most successful method because it presented the least number of frauds misclassified as false positives and false negatives. The study avoids evaluating the methods with other performance metrics because it states that each of the three models was trained differently, and therefore cannot be compared with each other.

The objective of Al-Shabi (2019) was to build an Autoencoder model to detect credit-card frauds on an European transaction database. This database is the same as the work of Awoyemi et al. (2017), containing 284,807 records with 492 fraudulent transactions. The database is used in its original form without applying any sampling technique, being divided into 80% for training and 20% for testing the model. At the end of the experiment, the authors define a construction error rate threshold that will define a transaction as fraudulent. The Autoencoder is evaluated with four different threshold values, and the authors conclude that the ideal value should be a balance between detecting more true frauds (higher accuracy) while keeping an acceptable value of false positives cases. The best results for the threshold of 5 are Accuracy (0.98), Precision (0.011), Recall (0.64), and F1-Score (0.19).

The study of Misra et al. (2020) features a two-stage fraud detection study using an unsupervised Autoencoder model combined with a supervised classification model. The database is the same as the work of Al-Shabi (2019) and Awoyemi et al. (2017), containing 0.17% of transactions labeled as frauds. The Autoencoder is applied as a first stage for detecting and extracting the main features of the database, generating a new database with fewer but more significant features. This reduced database is then submitted to the second stage, which is the classification process with three different models previously trained: Multilayered Perceptron (MLP), K-nearest Neighbor (KNN), and Logistic Regression (LR). The performance metrics used in the study are Accuracy, Precision, Recall, and F1-Score, the latter being the most important according to the authors. The conclusion of the study is that the application of Autoencoder combined with the MLP classifier achieved the best F1-Score (0.8265), Accuracy (0.9994) and Precision (0.8534) values when compared to the others.

3. DATASET AND PERFORMANCE EVALUATION

3.1 Database

The database is a proprietary real-world dataset provided by a Brazilian fintech and contains credit-card transactions collected from the company's cardholders during the entire year 2019. The database contains 39,465,007 credit-card transactions, of which 39,443,703 are legitimate and 21,304 are considered fraud. The amount of fraud represents only 0.053% of the total transactions, which means we need to deal with an unbalanced database. Table 1 presents the database attributes. The database has only numeric attributes, 12 in total.

Table 1. Database attributes

Attribute	Description
Amount	Purchase amount in Brazilian real
Average ticket	Average customer purchase amount at the time of the transaction, in Brazilian real
Hour	Purchase hour (numerical 0-23)
Day of week	Purchase day of week (numerical 0-6)
Currency	ISO 4217 code of purchase currency
Online	Flag indicating whether the purchase was face-to-face or not
First purchase	Flag indicating if is the customer's first purchase
Installments	Purchase installment: if the purchase amount was divided into several invoices. The minimum installment value is 1
Limit	Customer's card limit at the time of purchase
Quantity	Number of transactions from the same customer in the last 10 minutes
Merch. cat. code	Merchant type code
Class	Purchase class, if it is legitimate or fraud

It is worth mentioning that the database transactions are previously analyzed through the outsourced fraud detection systems of the acquirer and the credit-card network. The system that already exists in the acquirer analyzes simple rules, such as checking the time of the transaction (if done at dawn, for example, they are more suspicious), if they are made online or face-to-face, the country where it occurred, and if the password was required. Customer data is not analyzed by the acquirer for privacy reasons, as well as data on past transactions. The rules are fixed and applied to any purchase. The systems rely on a person who observes the rules and adjusts them from time to time. The system used by the credit-card network, if any, is a black box and its rules are not disclosed. Thus, the system we developed in this paper deals only with the most complex frauds, which were not previously detected by the fraud systems of the acquirer and the credit-card network.

Another challenge is that the transactions are possibly labeled as frauds through customer contact when reporting unrecognized purchases. The transaction is only confirmed and labeled as fraud after such contact and after a manual analysis by the fintech responsible area. Hence, there may be legitimate transactions that are frauds but have not been labeled as such because it was not correctly reported by the customer.

The database file is loaded into the algorithm and used in its original form without any previous manipulation. A check performed after the data loading ensures that the database is complete, with no missing records. The database also does not contain features in an invalid or null state. All numeric attributes are submitted to a scale function as they have different scales and magnitude. This process is known as feature scaling or data normalization. The database features are resized to have the properties of a normal distribution, with a mean of 0 and a standard deviation of 1.

The split to generate the train and test subsets is made using the Month attribute. The transactions from January to October (values 1 to 10) are separated for training and the months of November and December (values 11 and 12, respectively) are separated for testing the model. As this attribute was used to split the records, it is not used in the model evaluation. This strategy was chosen because it better simulates the real world. The months of November and December, in this strategy, simulate the new transactions and frauds occurring over time, using only the past records as training data.

All fraud records (where the Class attribute is equal to 1) are removed from the training subset because predictive models for anomaly detection should be trained only with data from the majority class, which are the legitimate transactions. At the end of this process, the training and testing subsets have 30,449,205 and 8,999,221 records respectively.

3.2 Performance evaluation

The following evaluation metrics are Accuracy, Recall, and Specificity. Their result numbers on a scale from 0.0 to 1.0, where 1 represents excellence in prediction. The metrics equations are:

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

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

$$Specificity = \frac{TN}{(TN + FP)} \tag{3}$$

where the results of the predictive model are true positive (TP) refers to the amount of fraud properly classified, false negative (FN) refers to the amount of fraud classified as legitimate, false positive (FP) refers to the number of legitimate transactions classified as fraud, and true negative (TN) refers to the number of legitimate transactions correctly classified. Precision is the fraction of true frauds among all samples which are classified as frauds, while recall is the fraction of frauds that have been classified correctly over the total amount of frauds.

As this study uses an unbalanced database, the metrics described above are not enough to reflect the model performance. As the number of legitimate transactions is 99.94% of the total, a model that, for example, mistakenly classifies all fraudulent transactions as legitimate produces a very high accuracy value. It may lead us to a false impression that the model has a good performance when it is not detecting any fraud at all. Because of this, two other

metrics are computed, Matthews Correlation Coefficient (MCC) and Area Under the ROC Curve (AUC), which are used for performance analysis as they are not affected by the class imbalance. These metrics were used in the research of Awoyemi et al. (2017) and Bhattacharyya et al. (2011) given their relevance in evaluating binary classification problems on unbalanced databases. The equation of MCC is:

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (4)$$

The result of the MCC is a number between -1.0 and +1.0. A value of +1.0 represents an excellent classification and, the value of -1.0 represents a distinction between classification and prediction.

The AUC Score is calculated by the two-dimensional area underneath the ROC curve (Receiver Operating Characteristic curve). The ROC curve graph plots the False Positive Rate on the X-axis and, the True Positive Rate on the Y-axis at different classification thresholds. The ideal point of the ROC curve is the top left corner of the graph as this maximizes the area under the curve. The result of the AUC Score is a value between 0 and 1. An excellent model has an AUC Score close to 1, which means that it has a good measure of classes' separability. The worst model has an AUC Score equals 0. When the AUC Score is 0.5 it means that the model has no class separation capability at all.

4. AUTOENCODER MODEL

An Autoencoder is a type of artificial neural network that learns to copy its input to its output. It has an internal (hidden) layer that describes a code used to represent the input, and it is constituted by two main parts: an encoder that maps the input into the code, and a decoder that maps the code to a reconstruction of the input.

The Autoencoder architecture of this paper is represented by Figure 2. The size of the Autoencoder input and output layers is the same number of database features (11 neurons). The intermediate layers between the input and output, also called hidden layers, have fewer neurons than the number of features. This reduction (compress) allows the Autoencoder to extract the main features of the records and then recreate them again to achieve the same number of features in the output. The compress layers are called encoders and, recreation layers are called decoders. The model consists of two hidden layers in the encoder with 6 and 3 neurons, respectively. The decoder has the same number of neurons in a mirrored way to guarantee an output with the same number of features as the input.

It was necessary to define three more parameters to compile the Autoencoder: overall metric, loss metric, and optimization function. The overall metric chosen was accuracy. The loss metric is the mean squared error (MSE) (Pumsirirat and Yan, 2018). The optimization function was Adam, suitable for models that deal with large databases and features. The Autoencoder model has been developed with the Python 3 programming language on the Jupyter

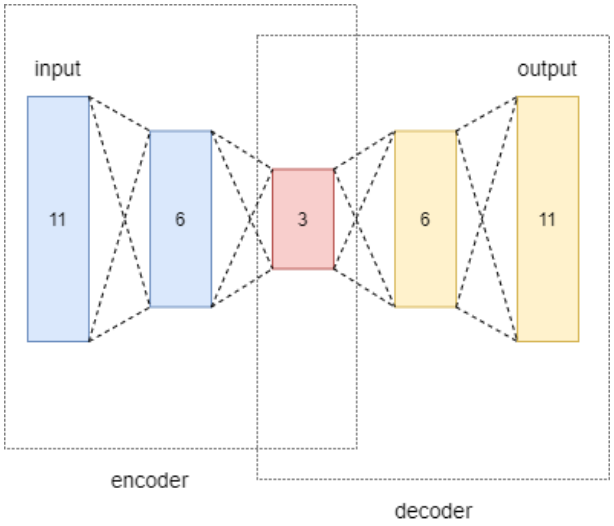

Figure 2. Autoencoder architecture.

Notebook platform. The neural network has been built using a model from the Keras framework with TensorFlow.

4.1 Autoencoder training

The Autoencoder model built was subjected to a training stage with 100 epochs and a batch size of 128 units. This allows the model to be trained in parallel using all available resources, gaining speed in processing databases with large amounts of data. During this process, the model performs an auto-evaluation in each step, optimizing itself based on the parameters defined in its initialization.

The calculation of the reconstruction loss defines the threshold that distinguishes a transaction from being an anomaly. The transactions that have a high rate of loss in the reconstruction process are considered anomalies because they are not equal to those already known by the model and they are classified as fraud. The lower the value of the mean squared error, the better is the model's result. The ideal MSE value is a number close to 0, which means a low loss in the reconstruction of a record by the Autoencoder. The accuracy curve and the loss rate for each epoch are illustrated by Figure 3 and Figure 4, respectively. It is possible to notice that the model increases the accuracy and reduces the MSE.

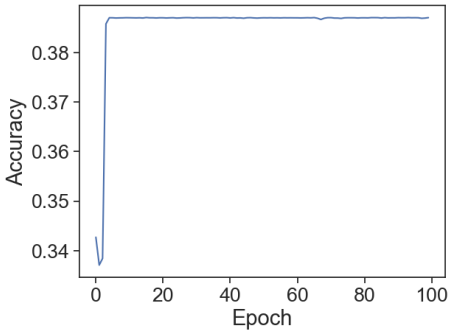

Figure 3. Autoencoder accuracy.

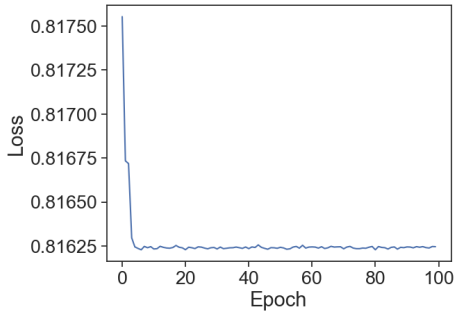

Figure 4. Autoencoder loss.

4.2 Autoencoder threshold definition

The following Figure 5 shows the specificity and recall values for different threshold values. The y-axis is not linearly showing the values, until the value 1, the divisions of 0.1 have the same distance from the divisions of 1, after the value 2. The graph shows the trade-off between the metrics, where while one metric increases, the other decreases. After analyzing the data, we chose the value 3 as the mean squared error threshold for separating the records. Such value maximizes the Specificity, and there is no significant increase after this point.

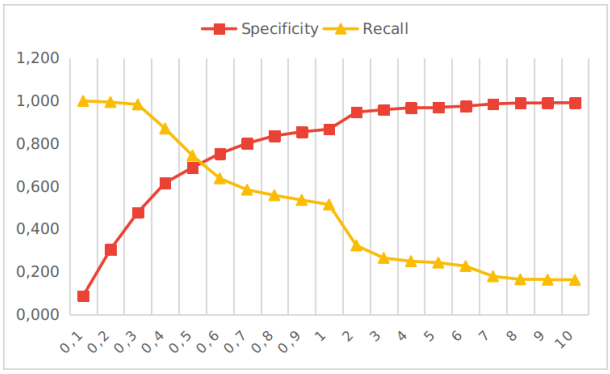

Figure 5. Specificity and recall values with various values of the threshold

Figure 6 illustrates the reconstruction error rate (MSE) of each record with the true class label from the test subset (legitimate or fraud). The classes are not linearly separable because many fraudulent transactions have reconstruction errors similar to legitimate ones. In addition, there are legitimate transactions with a high error rate, and some of them could be frauds but have not been reported by the cardholders yet.

5. EXPERIMENTS AND RESULTS

The training subset contains 30,449,205 records, all legitimate transactions because the model should be trained only with data from the majority class. The test subset contains 8,999,221 records with 8,994,498 legitimate and 4,723 fraudulent transactions (the fraudulent transactions are 0.05% of the total).

After training, the Autoencoder is subjected to the prediction stage using the test subset. The result of the model prediction is a matrix in the same format as the input subset, containing 8,999,221 reconstructed records with 11

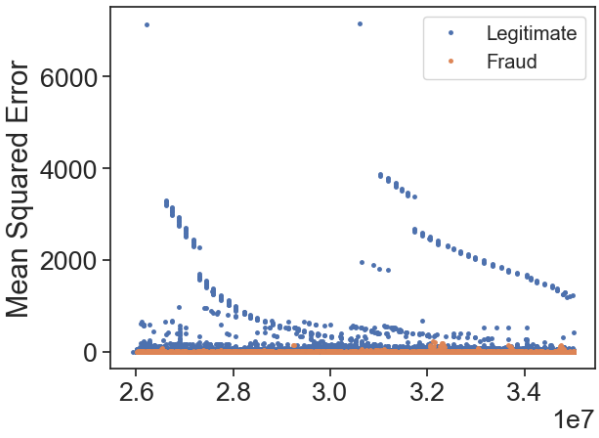

Figure 6. Mean squared error by class

features each. Then, the mean squared error is calculated for each of these records to find out how much loss there was in the reconstruction process. The classification of these records as anomalies and, therefore, fraud is done by the reconstruction error threshold that was defined in the previous section.

The confusion Matrix is illustrated by Figure 7, the ROC curve for calculating the area under the curve (AUC) in Figure 8, and the evaluation metrics are consolidated by Table 2.

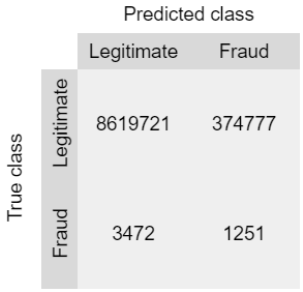

Figure 7. Autoencoder confusion matrix.

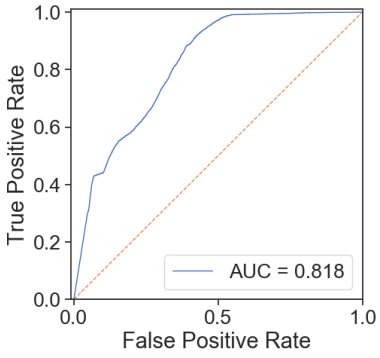

Figure 8. Autoencoder ROC curve.

Table 2. Autoencoder Metrics.

Metric	Value
Accuracy	0.95
Specificity	0.95
Recall	0.26
MCC	0.02
AUC	0.81

The Accuracy and Specificity metrics were 0.95, very close to 1 (the value of the best classification), because of the class imbalance. Therefore, both metrics can lead to a false impression of good performance. These two metrics use true negatives (legitimate transactions, the most represented class) as a numerator in their formula.

The Recall metric was 0.26, a lower value than the previous two metrics. This is because the model predicted more false negatives than true positives as seen by the confusion matrix, i.e., it classifies more fraudulent transactions as being legitimate than it can detect real frauds. Since real frauds are the least representative class in the database, this metric can also lead to a false impression of bad performance. It is possible to increase the number of true positive classifications and therefore improve the Recall metric, but this may also increase the number of false-positive classifications.

The AUC and MCC metrics are important for the continuity of Autoencoder performance analysis since the previous metrics are sensitive to class imbalance. A positive MCC value and an AUC value of 0.81 (greater than 0.5 and closer to 1) indicates that the Autoencoder model has a good overall capacity of class distinction and is not classifying in an inverse way to what was expected.

So, the Autoencoder model correctly classifies 95% of transactions, as shown by the Accuracy metric. However, for each wrong classification, there is a different cost from the credit-card issuer's point of view. This model would also approve 3,472 fraudulent transactions as being legitimate (false negatives). These transactions would generate a financial loss in any form even if there was no fraud detection system.

From the perspective of a credit-card issuer that decides to implement a fraud system with Autoencoder, its model would classify 1,251 transactions as being frauds correctly, and 374,777 wrongly. If the issuer decides to automatically block every transaction classified as fraud, there will be no financial loss of 1,251 transactions that would be fraudulent. But this block would also cover the other 374,777 legitimate transactions incorrectly classified as fraud (the false positives). This could cause customer frustration because these are legitimate purchases that are not being approved by the model.

If the credit-card issuer prefers to change the behavior and improve the classification to detect more real frauds transactions, the ROC graph analysis shows that there is a confidence threshold value where it is possible to increase the classification rate of true positives (i.e., increase the identification of real frauds) but at the cost of increasing the number of false positives, and as a result, this may increase customer frustrations. This trade-off must be carefully analyzed by the issuer as it directly affects the customer experience.

6. CONCLUSION

Credit-card fraud cases have become a recurring issue in recent years due to the growth of the internet and the ease of online shopping through e-commerce. Therefore, an Autoencoder was developed using Deep Learning for fraud detection in credit-card transactions. The Autoencoder we built for this paper was trained and validated using an original Brazilian database containing 39,465,007 credit-card transactions from the year 2019. After analyzing the reconstruction error rate of the test subset, a separation threshold was defined so that the records can be classified as legitimate or fraudulent transactions. The experiments presented satisfactory results showing that the Autoencoder model maintains the separability of classes even in a real-world Brazilian database, different from the works that use European databases. Besides, the trade-off of threshold tuning behaves in the same way.

A positive MCC value (0.02) and an AUC of 0.81, which are not affected by the database imbalance, also show that the Autoencoder has a greater capacity for class separation. It is worth mentioning that the database has only the most challenging frauds, which were not detected previously by the acquirer and the credit-card network fraud detection systems. Also, the result is promising considering that the database may have transactions that are actually frauds since the transaction labeling is made through customer contact reporting unrecognized purchases and after a manual analysis by the fintech responsible area.

Although the result of the experiment was successful in classifying transactions that were true frauds from the test subset, there were still transactions that were legitimate but were wrongly classified as fraud (false positives) by the Autoencoder. Transactions in this category can generate frustration in customers' experience, as they would have their purchases denied if the credit-card issuer decided to implement a system that automatically blocks all transactions classified as fraud.

The credit-card issuer must evaluate the cost of introducing a fraud detection system that is performative but can in some way impact the customer experience. Reducing the Autoencoder threshold value to increase the detection of real fraudulent transactions (true positives) is a trade-off in relation to the experience, as it also increases the number of false-positive cases.

The database publication is planned for future work so it can be used by and then compared with other works of literature. Our focus in this paper was to find a suitable Autoencoder threshold that provides a good classification of fraudulent transactions. The database will be published anonymously for data confidentiality since the company does not allow the disclosure of its name.

An improvement for future work would be to combine different Deep Learning algorithms through Ensemble learning (Sohony et al., 2018) seeking to reduce the rate of false positives to improve the customer experience, without penalizing the detection of true positives (the real frauds). Another improvement suggestion would be to implement confidence levels in the Autoencoder reconstruction error rates so that the classification is no longer binary with a fixed value threshold. With this improvement, each confidence level can lead to different actions such as, for example, requiring verification by the cardholder by sending a text message to confirm or cancel the purchase, anticipating a possible fraud.

REFERENCES

Al-Shabi, M. (2019). Credit card fraud detection using autoencoder model in unbalanced datasets. *Journal of Advances in Mathematics and Computer Science*, 1–16.

Awoyemi, J.O., Adetunmbi, A.O., and Oluwadare, S.A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 1–9. IEEE.

Bhattacharyya, S., Jha, S., Tharakunnel, K., and Westland, J.C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), 602–613.

Breslow, S., Hagstroem, M., Mikkelsen, D., and Robu, K. (2017). The new frontier in anti–money laundering. *McKinsey & Company, New York, NY, USA, Nov.*

Fiore, U., De Santis, A., Perla, F., Zanetti, P., and Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 479, 448–455.

Knieff, B. (2016). Global consumer card fraud: Where card fraud is coming from. Technical report, Technical Report, Aite Group, Boston, USA. URL: www. aciworldwide. com.

Misra, S., Thakur, S., Ghosh, M., and Saha, S.K. (2020). An autoencoder based model for detecting fraudulent credit card transaction. *Procedia Computer Science*, 167, 254–262.

Pumsirirat, A. and 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*, 9(1), 18–25.

Rezapour, M. (2019). Anomaly detection using unsupervised methods: credit card fraud case study. *Int J Adv Comput Sci Appl*, 10(11).

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., and Beling, P. (2018). Deep learning detecting fraud in credit card transactions. In *2018 Systems and Information Engineering Design Symposium (SIEDS)*, 129–134. IEEE.

Sohony, I., Pratap, R., and Nambiar, U. (2018). Ensemble learning for credit card fraud detection. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 289–294.

Zhang, J., Gardner, R., and Vukotic, I. (2019). Anomaly detection in wide area network meshes using two machine learning algorithms. *Future Generation Computer Systems*, 93, 418–426.