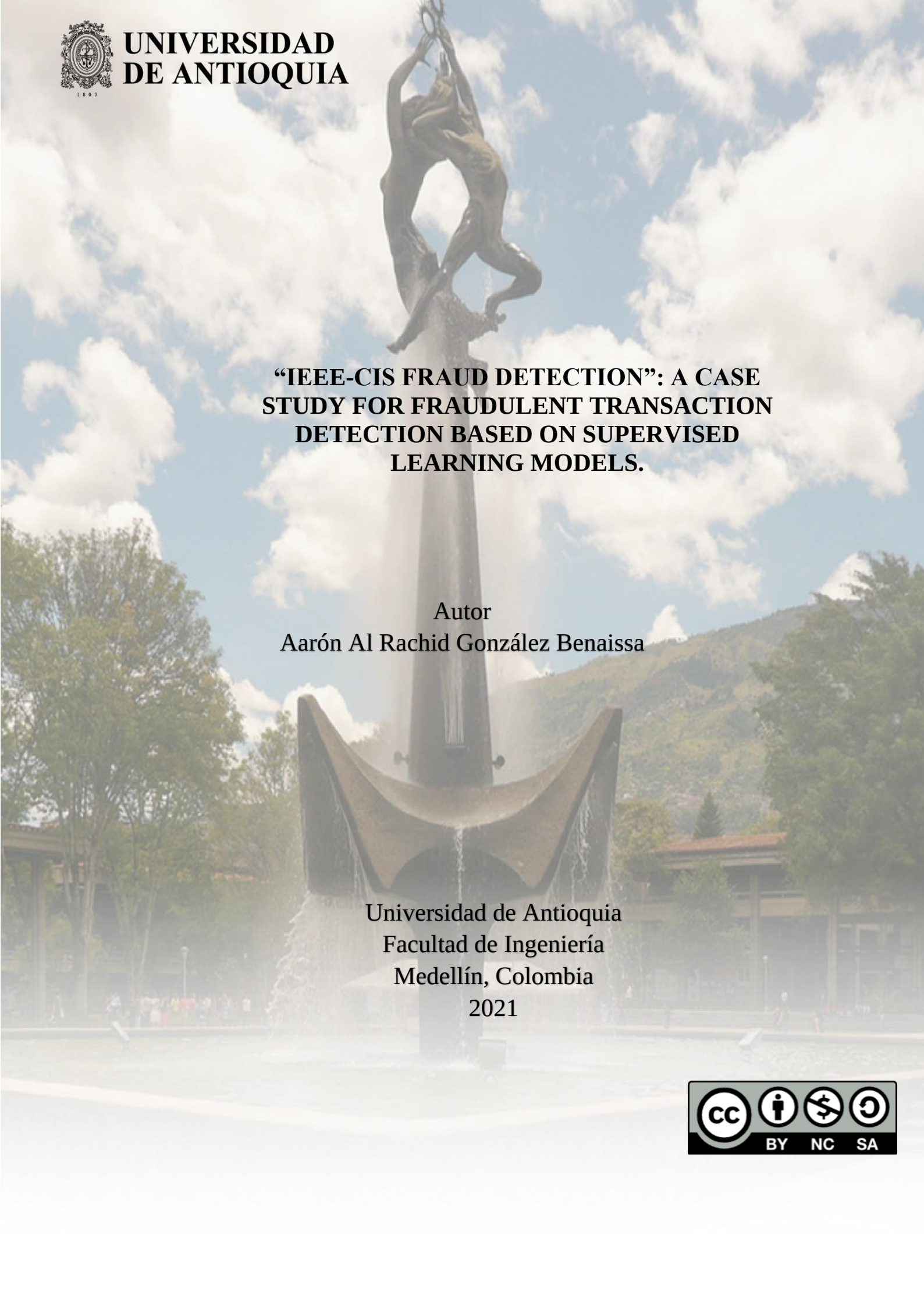

UNIVERSIDAD DE ANTIOQUIA

“IEEE-CIS FRAUD DETECTION”: A CASE STUDY FOR FRAUDULENT TRANSACTION DETECTION BASED ON SUPERVISED LEARNING MODELS.

Autor
Aarón Al Rachid González Benaissa

Universidad de Antioquia
Facultad de Ingeniería
Medellín, Colombia

2021

“IEEE-CIS Fraud Detection”: a case study for Fraudulent transaction detection based on Supervised Learning Models.

Aarón Al Rachid González Benaissa

Trabajo de investigación presentado como requisito parcial para optar al título de:
Especialista en Analítica y Ciencia de Datos

Asesora:
Lina María Sepúlveda Cano

Línea de Investigación:
Detección de anomalías haciendo uso de técnicas de Machine Learning

Universidad de Antioquia
Facultad de Ingeniería
Medellín, Colombia
2021.

TABLA DE CONTENIDO

1. INTRODUCTION 5

2. DATASET SPECS 5

3. DATA PREPARATION 6

4. EXPLORATORY DATA ANALYSIS 6

4.1. Categorical variables 6

4.2. Continuous variables 6

5. PREPROCESSING 7

5.1. Encoding categorial features: 7

5.2. Split and scaling data: 7

5.3. PCA decomposition 7

6. MODEL IMPLEMENTATION 7

7. CONCLUSIONS 7

8. REFERENCES 8

LISTA DE TABLAS

Table 1. Transaction table..... 5

Table 2. Identity Table..... 6

Table 3. Dataset target - legitimate and fraud transactions..... 6

Table 4. Confusion matrix for classifiers output 7

Table 5. Evaluation metrics for each classifier..... 7

“IEEE-CIS Fraud Detection”: a case study for Fraudulent transaction detection based on Supervised Learning Models.

Aarón Al Rachid González Benaissa

**Universidad de Antioquia, Facultad de Ingeniería*
Colombia (e-mail: aaron.gonzalez@ udea.edu.co).

Link: https://github.com/AaronGonzalezB/monografia-especializacion-udea.git

Abstract: This paper proposes a solution to the Kaggle competition: *"IEE-Fraud Detection"*, whose objective is to detect fraudulent transactions in a customer and transactional dataset collected by an E-commerce site to construct a transaction confirmation system via text messaging of the payment services company *Vesta Corporation*. Exploratory analysis of the data and different modeling approaches are shown, selecting the most appropriate results for anomaly detection.

Keywords: Fraud detection, binary classification, imbalanced data, dimensionality reduction

1. INTRODUCTION

With the auge of virtual transactions, billions of transactions are generated every day worldwide, this represents to financial companies a big challenge to maintain a secure platform to guarantee secure transactions, because of this, a Machine Learning framework are used to classify anomalies in transactions to protect the customers from digital scammers [1].

Classification algorithms play an important role in anomaly detection. At the financial level, it is possible to model their behavior and predict whether a transaction is like those previously identified as anomalous and classify it as fraudulent, if it is like valid transactions, classify it as a valid transaction and thus identify the types of transactions that are made on any user interaction platform.

Supervised learning algorithms such as decision trees, random forests [2], ensemble trees and from the neural networks approach Autoencoders [3] are suitable under a hyperparameter configuration that supports the unbalanced output variable.

In this paper, a binary classification algorithm is shown, where a transaction can be classified as valid or fraudulent depends on the data behavior. This problem can be worked as a Supervised Learning Algorithm because is a marked dataset. The transaction specs are collected by the security company *Vesta Corporation* [4], they bring the masked dataset to protect the user’s privacy.

The paper contains a detailed description of the dataset, a data preparation section combined with Exploratory Data Analysis (EDA) and the construction of three Supervised Learning Models: Logistic Regression, Random Forest, and a Lightgbm, all of them pass through a grid search to find the better estimators and metrics due to the imbalanced data, the metrics evaluated are the Confusion Matrix and the ROC Curve (Kaggle submission for this competition) [4].

2. DATASET SPECS

The data contain 393 features and 5.9 million transactions and is broken into two files: “identity”, for the session open that make the transaction and “transaction” that corresponds to the detail of the movement [4]. Not all transactions have corresponding identity information, but it can be joined by “TransactionID”.

Features of transactional information are shown in *Table 1*:

Feature	Description
TransactionDT	timedelta from a given reference datetime.
TransactionAMT	transaction payment amount in USD.
ProductCD	product code, the product for each transaction.
card1 - card6	payment card information, such as card type, card category, issue bank, country, etc.
addr	billing country (addr2) and billing region (addr1).
dist	distance.
P_ and (R__) emaildomain	purchaser and recipient email domain.
C1-C14	counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.
D1-D15	timedelta, such as days between previous transaction, etc.
M1-M9	match, such as names on card and address, etc.
Vxxx	Vesta engineered rich features, including ranking, counting, and other entity relations.

Table 1. Transaction table.

Session information are shown in *Table 2*, it contains the identity, network connection and browser details of the opened session for each transaction:

Feature	***Description***
TransactionID	*ID of transaction*
DeviceType	*device type entered.*
DeviceInfo	*device information.*
id_12 - id_38	*masked features corresponding to the detail of the open session and user logging.*

Table 2. Identity Table

All these variables are collected by Vesta's fraud protection system and digital security partners (the field names are masked and pairwise dictionary will not be provided for privacy protection and contract agreement).

The data was collected in production in a real environment, the target *isFraud = 1* was labelled based on the client reports of the card transactions, for the fraudulent transactions that never been reported, these could have been labeled as legit (*isFraud = 0*). The dataset is unbalanced based on the label of the transaction, this label define wether or not if a transaction is fraud (1) or legit (0), the ratio of this target variable is shown in *Table 3*:

isFraud	***% data***
0	96.5
1	3.5

Table 3. Dataset target - legitimate and fraud transactions

This behavior defines the basis for the exploratory data analysis because the weight of fraudulent must be balanced against the weight of legit transactions and the metrics must be balanced to avoid bias. This means we have to adapt some hyperparameters configuration in the search of the best estimator for each model, a grid search approach can be helpful to find the best model configuration.

3. DATA PREPARATION

In this analysis the transaction table and identity table were joined to make a unique dataset with *transactionID* key to have knowledge about the complete transaction and their missing data.

To clean the data before the variable transformations, a variable imputation was established, where the features with more than 80% of missing values are considering non-informative for the analysis. A total of 50 variables were deleted.

To better handle the data and optimize the execution memory the data types of the variables are cast to a lighter data type in some parts of the process and a garbage collector to free up memory capacity.

4. EXPLORATORY DATA ANALYSIS

4.1. Categorical variables

With a descriptive visualization, we can estimate the incidence of certain variables in case they are related to the target.

- *Target distribution per ProductCD* (*Figure 1*), referring to the buyed product code.

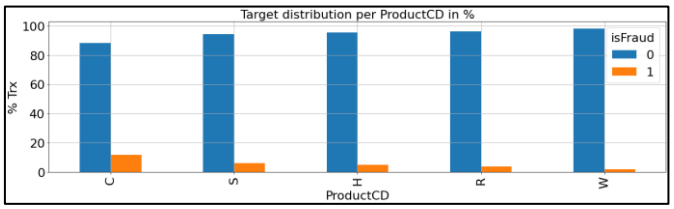

Figure 1. Target distribution per ProductCD

- *Target distribution per card4 (Figure 2)*, corresponds to the Credit Card franchise used in the transactions.

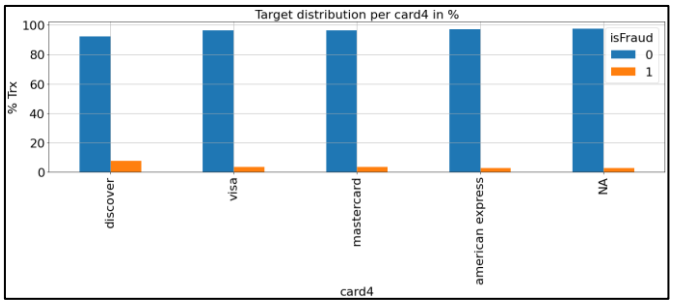

Figure 2. Target distribution per card4.

- Target distribution per card6 *(Figure 3)*, corresponds to the type of card used in the transaction.

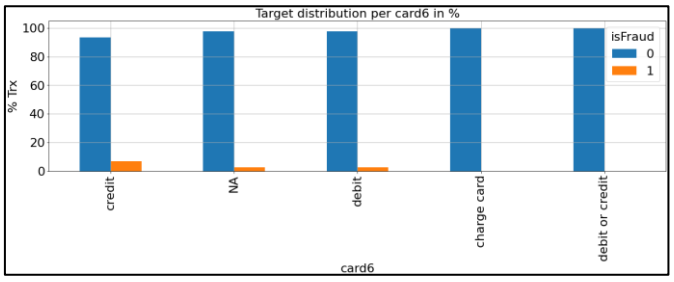

Figure 3. Target distribution per card6.

i. *Charge card*: special type of card, full payments month to month, no limit on quota, no interest charged.
ii. *Credit*: Credit card.
iii. *Debit*: Debit card.
iv. *Debit or credit*: there is not enough information to identify the exact type.

4.2. Continuous variables

To achieve a better understanding of the behavior of the continuous variables, some statistics such as means, medians, standard deviations and Pearson and Spearman correlation

matrices were implemented by groups to determine their correlation and incidence when performing the dimensionality reduction process.

In this case the variables provided by Vesta showed a high correlation (above 95%) and a high variance between them, so it is inferred that they come from a previous preprocessing process which is unknown. For this reason, a dimensionality reduction process was performed to contemplate all the variables, as will be seen in the next section.

5. PREPROCESSING

5.1. Encoding categorial features:

Using the *One Hot Encoding* approach to transform categorical into numerical variables to handle the scikit-learn models [5], then the dataset transformed has now 455 variables ready to scale.

5.2. Split and scaling data:

With a train/test split on the joined data with a test size of 20% stratified by the target variable, a *Robust Scaler* approach was used to scale the data through the median and the interquartile range, with this method the remaining outliers have less influence over other scaling methods [5].

5.3. PCA decomposition

To handle the large number of features after the One Hot Encoding transformation, it was necessary to make dimensionality reduction with PCA [6] and obtain the principal components over the 455 features. With this approach a cumulative variance evaluation was evaluated to obtain the number of features that covering more than 90% of the dataset variance, thus avoid losing relevant information. A total of 70 principal components was selected with this technique with a cumulative variance of 91%.

6. MODEL IMPLEMENTATION

With a grid search hyperparameter, some classification models were evaluated, first with a baseline Logistic Regression [7], then a Random Forest [2], and finally with Lightgbm approach to increase the complexity.

All models were trained with a grid of hyperparameters over a grid search, set for handle an unbalanced dataset. The metrics calculated over the test set were the False/True Positive and Negative rates, balanced accuracy score, precision score and recall, finally an AUC Roc curve was detailed to explain each model.

The normalized confusion matrix is shown in *Table 4*, it provides the False Positive (FP), True Positive (TP), False Negative (FN) and True Negative (TN) rates for each classifier, starting with the Logistic Regression as a Baseline model with less TP rate than the Random Forest and Lightgbm results [8].

Classifier	***Actual***	***Predicted***	
		No Fraud	*Fraud*
Logistic Regression	*No Fraud*	*0.74*	*0.26*
	Fraud	*0.31*	*0.69*
Random Forest	*No Fraud*	*0.84*	*0.16*
	Fraud	*0.27*	*0.73*
Light GBM	*No Fraud*	*0.87*	*0.13*
	Fraud	*0.28*	*0.72*

Table 4. Confusion matrix for classifiers output

The detail of evaluation metrics to the test set are shown in *Table 5*, it contains:

- *Precision*: number of TP over the total positive classified instances. Closer to 1 the better the precision.
- *Recall*: umber of TP over the total positive instances. Closer to 1 the better the recall.
- *F1*: weighted average of the precision and recall, where the contribution of both measures is equal. Closer to 1 the better the F1.
- *MCC*: correlation measure between the observed and predicted two-class classifications. The value range is between -1 and +1, where 1 indicates best prediction and -1 otherwise.
- *Accuracy*: number of correct predictions over total predictions, for an unbalanced dataset it can generate the "accuracy trap", if both classes have the same weight, a good accuracy does not represent that it is classifying both classes correctly (since the majority class would take a greater number of instances). For this reason, balanced accuracy is recommended.
- *Balanced accuracy*: mean of sensitivity (TP) and specificity (TN). This avoids hiding the performance in predictions of minority class.
- *ROC AUC Score:* measure to define the relation between TP rate and FP rate.

Classifier	***Precision***	***Recall***	***F1***	***MCC***	***Accuracy***	***Balanced accuracy***	***AUC ROC Score***
Logistic Regression	*0.533*	*0.697*	*0.499*	*0.163*	*0.742*	*0.697*	*0.697*
Random Forest	*0.539*	*0.716*	*0.514*	*0.185*	*0.76*	*0.716*	*0.716*
Lightgbm	*0.594*	*0.789*	*0.627*	*0.331*	*0.895*	*0.789*	*0.789*

Table 5. Evaluation metrics for each classifier

7. CONCLUSIONS

In this work, a modeling and data treatment process was defined for a real transactional system that reflects the need to know the data to avoid fraudulent transactions and complement the early warning system defined in the competition. The information was explored in such a way that,

despite not being explicitly known, preprocessing and modeling techniques were used to identify its dynamics and prepare the data for an environment in accordance with Machine Learning models.

For this type of problems where the data are not explicitly known, the detailed exploration of a group of variables and their preprocessing, as well as simplification and codification techniques to efficiently manipulate the information and be able to execute it in a production environment, are of vital importance.

Additionally, it is necessary to explore models with a more complex configuration to increase the performance metrics, given the unbalance of the data with respect to the target variable, so in this case the Lighgbm model was the one that obtained the best results according to the metrics presented.

8. REFERENCES

[1] M. H. Khadija AbdulSattar, "Fraudulent Transaction Detection in FinTech using Machine Learning Algorithms," *International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, p. 1, 2020.

[2] G. W. Q. H. M. H. Y. Du Shaohui, "Customer Transaction Fraud Detection Using Random Forest," *IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2021.

[3] O. A. A. A. A. M. Y. Hassan Najadat, "Credit Card Fraud Detection Based on Machine and Deep Learning," *International Conference on Information and Communication Systems (ICICS)*, 2020.

[4] I. C. I. Society, "Kaggle - IEEE-CIS Fraud Detection," Vesta, 2019. [Online]. Available: https://www.kaggle.com/c/ieee-fraud-detection/. [Accessed 28 05 2021].

[5] "Scikit Learn API Reference - Preprocessing and Normalization," 2020. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing. [Accessed 28 05 2021].

[6] "Scikit learn API Reference - PCA," 2021. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition. [Accessed 28 05 2021].

[7] "Scikit Learn API Reference - Linear Models," 2021. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model. [Accessed 28 05 2021].

[8] "Lightgbm documentation," 2021. [Online]. Available: https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html. [Accessed 28 05 2021].

[9] "Scikit Learn API Reference - metrics," 2021. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics. [Accessed 28 05 2021].

[10] "Scikit Learn API Reference - Ensemble Methods," 2021. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble. [Accessed 28 05 2021].