Semi-supervised Method to Detect Fraudulent Transactions and Identify Fraud Types while Minimizing Mounting Costs

Chergui Hamza[1], Abrouk Lylia[2], Cullot Nadine[3], Cabioch Nicolas[4]
Université de Bourgogne, SKAIZen Group, 4 av. Alain Savary,
21000 Dijon[1,2,3]
SKAIZen Group, 14 rue de mantes,
92700 Colombes[4]

***Abstract*—Financial fraud is a complex problem faced by financial institutions, and existing fraud detection systems are often insufficient, resulting in significant financial losses. Researchers have proposed various machine learning-based techniques to enhance the performance of these systems. In this work, we present a semi-supervised approach to detect fraudulent transactions. First, we extract and select features, followed by the training of a binary classification model. Secondly, we apply a clustering algorithm to the fraudulent transactions and use the binary classification model with the SHAP framework to analyze the clusters and associate them with a particular fraud type. Finally, we present an algorithm to detect and assign a fraud type by leveraging a multi-fraud classification model. To minimize the mounting cost of the model, we propose an algorithm to choose an optimal threshold that can detect fraudulent transactions. We work with experts to adapt a risk cost matrix to estimate the mounting cost of the model. This risk cost matrix takes into account the cost of missing fraudulent transactions and the cost of incorrectly flagging a legitimate transaction as fraudulent. In our experiments on a real dataset, our approach achieved high accuracy in detecting fraudulent transactions, with the added benefit of identifying the fraud type, which can help financial institutions better understand and combat fraudulent activities. Overall, our approach offers a comprehensive and efficient solution to financial fraud detection, and our results demonstrate its effectiveness in reducing financial losses for financial institutions.**

Keywords*—Machine learning; semi-supervised learning; fraud; finance; cost analysis

I. Introduction

Financial institutions face multiple challenges in fighting money laundering activities. Jensen [1] defines it as *the disguise of the origin of illegally obtained funds to make them appear legitimate. The goal of money laundering is to convert cash into another form*. Financial institutions must fight fraudulent activities by analyzing their customers' transactions. Transactions exchanged between financial institutions use SWIFT (Society for Worldwide Interbank Financial Telecommunication). This provides an interbank network offering different services, such as money transfers between bank accounts. More than 11,000 banking organizations across nearly 200 countries use SWIFT to transfer money [2]. SWIFT transactions are international and may have multiple intermediaries between the transaction's originator and beneficiary. However, among these transactions, anomalies may be linked to financial fraud. Thus, the analysis of interbank transactions is a crucial issue for financial institutions [3]. The current systems have four limitations outlined by SWIFT[1]: 1) **systems and processes inefficiency**, originally designed for retail banking, is based on rules and risks. In this context, a critical problem is the high alert volume generated by them, and 90 percent of them are false alerts. Therefore, a manual investigation by experts is required. 2) Due to **mounting costs**, financial institutions lose a considerable amount of money fighting against money laundering—either by being fined for their weak compliance systems, maintaining these, or paying experts to review the alerts. 3) **Indirect structure**: Some domestic and regional banks act as aggregators for smaller banks. It is hard to follow and monitor the payment activity effectively in such instances. 4) **Information sharing**: Banks do not share information with their customers due to confidentiality clauses. In addition, due to these limitations, fraudulent transactions' detection is a complex task. It's also difficult to identify the fraud types [4]. For these reasons, our work aims to respond to the first three challenges by answering the following questions: How can fraudulent transactions be detected with their frauds types while minimizing the mounting costs ? The article is structured as follows: in Section II, we review the most recent developments in the field of machine learning for detecting financial fraud and identifying different types of fraud. Then, in Section III, we present our method, which aims to detect fraudulent transactions within the SWIFT network and identify fraud types, while minimizing the associated costs. In Section IV, we apply our methodology to a real-world dataset and report on the experimental results, demonstrating the effectiveness of our approach. Finally, in Section VI, we summarize our findings and outline potential avenues for future research.

II. Related Work

Most of the detection systems are based on rules. These systems, based on predefined rules pertaining to amounts, countries, or customer behaviors, are identified by fraudsters. In turn, they adapt their behaviors to bypass these rules and manage to launder their money through illegal activities. For these reasons, they generate a high false alerts rate and detect few fraudulent transactions. Most of the literature on

[1] https://www.swift.com/fr/node/166756

financial fraud detection techniques are based on machine learning. These techniques “offer numerical power and functional flexibility needed to identify complex patterns” [5]. Machine learning techniques for detecting financial fraud follow a process that can be divided into four steps: 1) **data acquisition**, which can be a complicated task for specific fields, such as finance or medicine, in instances where the data is confidential. Synthetic data can be used to validate experiments. 2) **Features extraction** involves calculating new information from existing features and reducing the number of dimensions while storing the information in the initial features. The 3) choice of **algorithms and their hyperparameters** depends on the number of dimensions, the volume, and the nature of the data. 4) The last step is the model **evaluation**. The predictive models are evaluated with metrics based on correct and wrong predictions.

This section is structured as follows: we present the machine learning process used for fraudulent transaction detection. Then we present a fraud types identification work. Finally, we synthesize the related work and introduce our approach.

A. Machine Learning Process

1) Data acquisition: A financial dataset is transactions with fields such as the amount, the date, the beneficiary [6], [7], [8]. They are confidential, and the lack of public data hinders experiments, particularly for their validation and comparison. There are different data formats depending on the data source (retail bank [9], agricultural bank [10]). The data volume in various experiments is heterogeneous, ranging from thousands [11] to millions [12] transactions. Validation of fraud detection approaches requires labeled datasets with legitimate and fraudulent transactions. A common aspect of financial fraud datasets is the class imbalance between fraudulent and legitimate transactions, with a fraudulent ratio usually around 0.1% [13]. In this context, researchers are interested on synthetic financial data generation. Lopez and al. [14] developed *PaySim* a mobile payment simulator tool with fraudulent transactions.

Most of the studied approaches don’t explain the data generation process [15], [10]. Michalak and al. [16] detail in their study how they generated transactions. They use Gaussian distribution to generate a transactions networks between employees of companies. LV and al. [17] use real data coupled with artificially generated fraudulent data. Some approaches use the *kaggle* public dataset[2] to validate anti-money laundering methods [18], [7], [19]. This dataset comprises transactions conducted with credit cards involved in fraud. It includes the following three known attributes: the date, the amount, the transaction class (fraudulent or legitimate), and 28 remaining attributes with unknown meanings.

2) Features extraction: The data must be processed before training the models with algorithms. There are many types of processing, namely standardization, features addition, or dimensions reduction. Features from the transaction attributes are extracted to represent customers’ behaviors based on their transaction history. The transactions are aggregated with different periods (weekly, monthly, and yearly) to compute several features, such as the transactions’ average amounts made by customers and their frequency (number of transactions done in a period). There are few works based on SWIFT transactions [20] or international transactions fields (countries or currencies) [21] to extract features. When the features number is too high, a dimensions reduction step is used to train the models faster or for visualization purposes. Bestami et al. [19] use the PCA (Principal Component Analysis) algorithm to reduce dimensions number to train a model with the K nearest neighbors algorithm. Paula et al. [22] use auto-encoders (deep learning technique) to reduce dataset dimensions number and complete the training 20 times faster. As mentioned, financial fraud datasets are imbalanced. The class imbalance problem can be less constraining using over or under-sampling techniques. Oversampling techniques generate synthetic new instances from the minority class, whereas under-sampling is used to reduce the number of instances from the majority class. SMOTE [23] is a popular oversampling algorithm in the literature that generate additional fraudulent transactions from the existing ones in the dataset. Badal et al. [24] prove this technique effective in financial fraud detection by obtaining better results using the SMOTE algorithm.

3) Algorithms and hyperparameters: Fraud analysts use fraud detection rules. These rules are used to detect fraudulent scenarios that occur frequently. However, rules can become quickly obsolete and must be reviewed. Nowadays, machine learning can be combined with a rules-based system. Classification (supervised learning) and clustering (unsupervised learning) help in fraud prevention and detection by classifying transactions as fraudulent or legitimate. The choice between supervised and unsupervised learning depends on the datasets. Supervised learning aims to learn the relationship between the data and its label. In unsupervised learning, the goal is to retrieve exploratory information, by grouping similar data or detecting hidden patterns [5]. Ryman-Tubb et al. [25] conduct a survey on card fraud detection methods for using financial transactions. This survey shows that only eight methods can be deployed on real data. Al-Hashedi et al. [26] expanded the work of Albashrawi [27] and present a survey from 2009 to 2019 conducted on financial fraud classified by fraud types.

a) Supervised methods: These compare algorithms to deduce which is pertinent to the data and their volume [28], [29], [19]. Mehbodniya et al. [30] propose financial fraud detection in healthcare based on machine learning and deep learning techniques and showed that the KNN algorithm generates better results than other approaches. Ensemble methods such as random forest [31] or boosting algorithms [32], [33], which combine multiple models, also proved their effectiveness in imbalanced datasets by using local decisions taken in areas where the imbalance is less prominent.

b) Unsupervised methods: These are used for financial fraud detection. Porwal et al. [7] use K-means algorithm to create fraudulent and legitimate transaction clusters. Simultaneously, other works propose new distance measurements to detect outliers [34]. Guo et al. [35] use autoencoders to have a deep representation of their data; they combined it with a KNN-based outlier detection method [36].

c) Semi-supervised methods: This combines supervised and unsupervised learning. Some approaches [37], [38], [8], [39] apply unsupervised algorithms to label their data, then they use supervised algorithms to classify their transactions. These approaches don’t need an expert to verify each abnormal

[2] https://www.kaggle.com/mlg-ulb/creditcardfraud

transaction from the unsupervised model, they verify clusters and fraudulent transactions from the supervised model.

4) Evaluation and metrics: The trained models are evaluated to check their effectiveness. Many metrics for evaluation, such as precision, recall, or F1-Score, exist. These metrics are used for model validation and comparison. In the evaluation process, data volume and fraudulent class rate are important. The evaluation should be done on the minority class.

In the case of unlabeled datasets, the evaluation is more complex. The verification of prediction results can be a long and imprecise operation. Furthermore, in the financial domain, Bahnsen et al. [40] propose a cost–risk matrix (Table I) to estimate the model mounting cost for a financial institution fighting against credit card fraud. It estimates model mounting costs depending on its predictions: it has an administration cost C_a for transactions predicted fraudulent, representing the estimated price for a transaction investigated by an expert. Amt_i is a fraudulent transaction cost, the fraudulent transaction amount predicted as legitimate by the model. They sum up all the transaction costs used in the evaluation phase to compute the model mounting costs.

TABLE I. BAHNSEN [40] RISK-COST MATRIX

		Reality (t_i)	
		Fraud	Legit
Prediction (t_i)	Fraud	C_a	C_a
	Legit	Amt_i	0

B. Fraud Types Identification

Desrousseaux et al. [41] present an approach to profile money laundering activities. They use the SOM (Self-Organizing Map) algorithm from an unlabeled transaction dataset to map its transactions into a two-dimensions matrix. SOM algorithm [42] is used as an unsupervised algorithm to cluster and visualize data with a two-dimensional map. The algorithm assign a new representation for each transaction. It uses the map node value associated to the transaction. Desrousseaux et al. use this node and its neighborhood as a new transaction representation. With it, they train a neural network called Fuzzy ART, which forms clusters with transactions. Finally, they combine these clusters with the map from the SOM algorithm, resulting in a map with different regions depending on the value of the node associated with a cluster. To interpret results, they use two methods: 1) They use the Fuzzy Art model weighted vector for each money laundering type and features distribution to retrieve the features with the highest weight. 2) They group transactions with the same type and use features distribution. This approach is interesting because no literature interprets fraud types in financial datasets. The choice of algorithms might be questionable because there are numerous clustering algorithms, such as k-means or BIRCH. The interpretation through the comparison of maps can be very complicated depending on the number of dataset features. Moreover, this approach relies on unlabeled datasets. It does not address the banks' concern of identifying the fraudulent patterns on their labeled dataset and the fraud types identification. While their interpretation of feature distribution is interesting, a new tool in the literature has been designed to interpret models, which could be helpful. In our work, we aim to extend this work, propose a method to detect fraudulent transactions with a reduced features number, and interpret the fraud types with interpretable tools.

TABLE II. EXAMPLE OF SWIFT MESSAGES OF THE DATASET

Originator	Intermediary	Beneficiary	Date	Currency	Amount	Class
BIC0FR01	BIC0IT01	BIC0FR02	210625	EUR	15006	L
BIC0US03	-	BIC0GB01	210625	GBP	33065	L
BIC0FR04	BIC0FR06	BIC0FR05	210626	EUR	100325	F

C. Synthesis

We studied the related works of fraudulent transactions detection and types identification. This problem has not been considered yet on the SWIFT transactions. As mentioned, SWIFT executes financial transactions between banks worldwide. These transactions have fields such as: country, currency and intermediary. SWIFT fraudulent transactions detection model is obtained through a comparative study of the supervised and unsupervised algorithms and their evaluation.

Desrousseaux et al. [41] present interesting results to identify the fraud types inside our datasets. However, this approach could be improved by considering labeled datasets and using model interpretation (SHAP). We aim to reduce our model mounting costs by using and expanding the risk–cost matrix from [40].

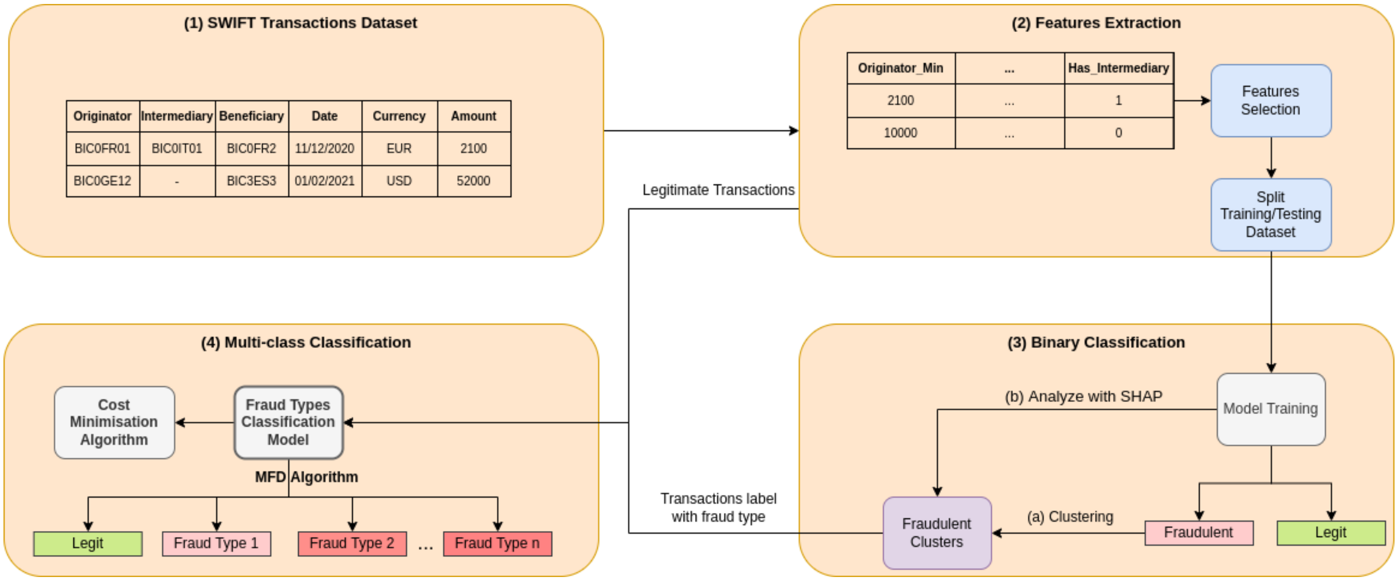
III. METHODOLOGY

In this section, we present our proposed approach for labeled transactions (fraudulent and legitimate). Our approach has three goals:

1) Detect fraudulent transactions
2) Identify and analyze the fraud types in the dataset
3) Minimize the mounting costs

To achieve these goals, we present our semi-supervised approach in Fig. 1. First, we extract features from a SWIFT transaction set. Second, we select the relevant features for detecting fraudulent transactions from the legitimate by training a binary classification model. Afterward, we create a fraudulent transaction subset, from which we apply an unsupervised algorithm to form clusters based on the fraud types. We also use the binary classification model to interpret the cluster to identify the fraud type. Moreover, we label fraudulent transactions based on the fraud types and train a multi-class classification model. We propose the *Multi-Fraud Detection* algorithm (MFD) to classify a transaction as legitimate or classes associated to fraud types. Then we propose a second algorithm to minimize the model mounting cost.

This section is structured as follows: (A) we present the dataset and the fields, (B) we list the extracted features, (C) we train a binary classification model and evaluate it, (D) we apply a clustering algorithm on the fraudulent transactions and identify the fraud type, (E) we train a multi-class classification model, and (F) we minimize the model mounting costs.


(1) SWIFT Transactions Dataset
(2) Features Extraction
Features Selection
Split Training/Testing Dataset
Legitimate Transactions
(4) Multi-class Classification
Cost Minimisation Algorithm
Fraud Types Classification Model
MFD Algorithm
Legit
Fraud Type 1
Fraud Type 2
...
Fraud Type n
Transactions label with fraud type
(3) Binary Classification
(b) Analyze with SHAP
Model Training
Fraudulent Clusters
(a) Clustering
Fraudulent
Legit

Originator	Intermediary	Beneficiary	Date	Currency	Amount
BICOFR01	BIC0IT01	BICOFR2	11/12/2020	EUR	2100
BIC0GE12	-	BIC3ES3	01/02/2021	USD	52000

Originator_Min	...	Has_Intermediary
2100	...	1
10000	...	0

Fig. 1. Methodology architecture schema

A. Dataset Presentation

In Table II, we present the SWIFT transactions fields. These transactions contain three actors: the originator, the intermediary, and the beneficiary. The transaction corresponds to the transfer of a money **amount** in a **currency** operated on a **date** between an **originator** and a **beneficiary**. SWIFT transactions path depends on the relationship between the **originator** bank and the **beneficiary** bank. If they have no direct relationship, then the transaction involves an **intermediary** connecting the two banks. Otherwise, the path only includes the originator and the beneficiary. An actor is identified with a BIC code, which contains the financial institution country. The transaction class indicates if the transaction is fraudulent (F) or legitimate (L). We formalize transactions as the following:

We have a set T of transactions t_i, where i ranges from 1 to n, the total number of transactions. To compute features for each transaction t_i, we form transactions subset with a similar field (e.g. originator), with specific time window, (e.g. last 15 days). For example, for a transaction t_1, we have subset $T^{15D}(originator)$ containing all transactions with the same originator for the last 15 days.

B. Detection of Fraudulent Transactions

We need to extract features from the dataset to separate legitimate and fraudulent transactions. The computed features must enlighten the fraud associated with SWIFT transactions. In a transaction, we have one numerical field: the amount. Features are computed on different period with the date field. Hence, we use the amount and the date to compute features associated with the other fields: the originator, the intermediary, the beneficiary and their countries, and the currency. With a financial expert, we define a list of features with their formalization in Table III.

We also define other features relative to the transaction path. In this context, by path, we mean the countries implied in the transaction and the presence of an intermediary. We list the features in Table IV.

Finally, we add temporal features relative to the day, the week, the year. We select a reduced features number. Some features might be irrelevant to the separate legitimate and fraudulent transactions in the features computed. A feature selection algorithm assigns an importance value to each feature. We select the top n features depending on the feature number required to interpret the fraud type. With the additional features, we train a binary classification model, evaluated with the metrics presented in Section III-C.

C. Model Training and Evaluation

In the related works, researchers trained a model with different classifiers and then selected the classifier with the best evaluation results. A good evaluation is crucial to ensure that the extracted features can distinguish between legitimate and fraudulent transactions. To do this, we use a confusion matrix presented in Table V. TP is the number of true positives. FN is the number of false negatives. FP is the number of false positives. Finally, TN is the number of true negatives.

From the confusion matrix, we can evaluate our approach with the classical metrics:

$$Precision = \frac{TP}{TP + FP}; \tag{1}$$

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

$$F1 = \frac{2 * TP}{2 * TP + FP + FN}; \tag{3}$$

D. Clustering of Fraudulent Transaction

Unsupervised algorithms are effective at discovering new patterns and hidden relations in datasets. Thus, we create

TABLE III. Fields based Features

Formalization	Description
$\lvert T(field)\rvert$	Transactions number in the set
$\underset{0<i<n}{max}\ T(field\mid amount)$	Maximum amount
$\underset{0<i<n}{min}\ T(field\mid amount)$	Minimum amount
$\sum_{i=0}^{t} T(field\mid amount)$	Sum of transactions
$\frac{\sum_{i=0}^{t} T(field\mid amount)}{\lvert T(field)\rvert}$	Average transactions amounts
Latency	Seconds since the last transaction
$\lvert T(field) \cap T(Originator)\rvert$	Count of transactions with the same Originator
$\lvert T(field) \cap T(Intermediary)\rvert$	Count of transactions with the same Intermediary
$\lvert T(field) \cap T(Beneficiary)\rvert$	Count of transactions with the same Beneficiary
$\lvert T(field) \cap T(Currency)\rvert$	Count of transactions with the same Currency
$\lvert T(field) \cap T(OriginatorCountry)\rvert$	Count of transactions with the same Originator Country
$\lvert T(field) \cap T(IntermediaryCountry)\rvert$	Count of transactions with the same Intermediary Country
$\lvert T(field) \cap T(BeneficiaryCountry)\rvert$	Count of transactions with the same Beneficiary Country
$\lvert\{T(field) \cap T(Originator)\}\rvert$	Distinct count of transactions with the same Originator
$\lvert\{T(field) \cap T(Intermediary)\}\rvert$	Distinct count of transactions with the same Intermediary
$\lvert\{T(field) \cap T(Beneficiary)\}\rvert$	Distinct count of transactions with the same Beneficiary
$\lvert\{T(field) \cap T(Currency)\}\rvert$	Distinct count of transactions with the same Currency
$\lvert\{T(field) \cap T(OriginatorCountry)\}\rvert$	Distinct count of transactions with the same Originator Country
$\lvert\{T(field) \cap T(IntermediaryCountry)\}\rvert$	Distinct count of transactions with the same Intermediary Country
$\lvert\{T(field) \cap T(BeneficiaryCountry)\}\rvert$	Distinct count of transactions with the same Beneficiary Distinct country

TABLE IV. Path Features

Features	Description
Intermediary	Boolean value to specify if there is an intermediary
Originator path	The count of the originator using the intermediary and beneficiary countries to make a transaction
Intermediary path	The count of the originator using the originator and beneficiary countries to make a transaction
Beneficiary path	The count of the originator using the originator and intermediary countries to make a transaction
Distinct originator Path	The count of a distinct path with an intermediary and beneficiary countries
Distinct intermediary Path	The count of a distinct path with an originator and a beneficiary countries
Distinct beneficiary path	The count of a distinct path with an originator and an intermediary countries

TABLE V. Confusion Matrix

	Predicted : Positive	**Predicted : Negative**
Actual : Positive	TP	FN
Actual : Negative	FP	TN

fraudulent transactions clusters, each of which will be associated with a fraud type. Cluster analysis allows experts to identify which fraud types clusters are associated based on their fraud knowledge. We use the SHAP framework [43] to facilitate cluster analysis and interpret model predictions. It assigns importance to each feature (Shapley values) for a prediction, depending on the feature's value. By leveraging this framework, we use visualization tools: *heatmap* and *beeswarm* for each cluster.

E. Multi-Class Model

After identifying the fraud types, we train a new model to predict the transactions class between legitimate and fraud types. Class numbers rely on the fraud types' numbers on the dataset. The new model attributes a probability to each class; however, the frauds' probability is split into different classes. Some fraudulent transactions' probability is divided into different fraudulent classes. The legitimate class could, in turn, take over them. For these reasons, we sum the fraudulent classes' probability p_i ($1 < i < n$, n number of fraudulent classes), we start at 1 because $i = 0$ represents the legitimate class. The transaction is considered fraudulent if the sum of p_i is above a defined threshold. The transaction's class will be the fraudulent with the highest probability. We resumed this on our *MFD* algorithm presented in the Algorithm 1.

F. Mounting Costs Minimization

To estimate the model mounting costs, we used the matrix of Bahnsen et al. [40], where we added a cost C_d for the legitimate transaction predicted as fraudulent, which estimates the dissatisfaction price of a customer whose transaction has been blocked. Indeed, Bahnsen et al. did not consider the wrong prediction cost for a fraudulent transaction. If a transaction is incorrectly blocked for a customer, then this one could be unsatisfied and result in losses for the financial institution.

We minimize the model's risk cost with our new risk–cost

Algorithm 1 MFD : Multi-Fraud Detection

```
T : set of transactions
m : classification model
threshold : fraudulent threshold
n : fraudulent classes number
for $t$ in $T$ do
    $p$ = model.predict_proba($t$)                ▷ list of $p_i$
    $sum = \sum_{i=1}^{n} p_i$
    if $sum > threshold$ then
        $fraudtype\_index = argmax(p)$
        $class\_list$.add($fraud[fraudtype\_index]$)
    else
        $class\_list$.add($legit$)
    end if
end for
return $class\_list$
```

TABLE VI. Adapted Risk-Cost Matrix

		Reality (t_i)	
		Fraud	Legit
Prediction (t_i)	Fraud	$C_a + C_d$	C_a
	Legit	Amt_i	0

matrix in Table VI and Algorithm 2. For that we use the *MFD* algorithm for each threshold value between 0 and 1 with a 0.01 step. Subsequently, with the model's prediction with these thresholds, we compute the model, then, we add f1-score and cost in list. Afterward, we retrieve the highest f1-score and check the closest f1-score with the lowest cost. Moreover, we return the threshold corresponding to this, and by doing so, we ensure to keep our model effective while reducing its mounting costs.

Algorithm 2 Cost Minimization

```
T : set of transactions
ca : administrative cost
cd : dissatisfaction costs
amt : transactions' amounts list
class_target : transactions' class
model : trained model
f1_list : f1-score list
costs_list : costs list
for $threshold \leftarrow 0$ to $1$ by $0.01$ do
    class_list = MFD (T, model, threshold)
    f1_list.add(compute_f1(class_list,class_target))
    costs_list.add(compute_costs(class_list,
class_target,ca,cd,amt))
end for
best_f1 = max(f1_list)
thresholds_around_best_f1 = around(best_f1, f1_list)
best_threshold = argmin_cost(costs_list, thresholds_around_best_f1)
return best_threshold
```

IV. Experimentation

Experiments were conducted with a 3676795 SWIFT transactions dataset obtained through a collaboration with the SKAIZen Group[3] company. We split the data into a training dataset composed of 294136 transactions with 10722 fraudulent transactions and a testing dataset composed of 735359 transactions with 2645 fraudulent transactions. Thereafter, we used the Jupyter[4] platform to develop the experimentation. We trained our models with the Scikit-Learn library.

TABLE VII. Base Fields Name

Originator
Intermediary
Beneficiary
Common history between Originator and Intermediary
Common history between Originator and Beneficiary
Common history between Intermediary and Intermediary
Currency
Originator country
Intermediary country
Beneficiary country

TABLE VIII. The 10 Features Selected

Features Name
Value
number with intermediary Beneficiary 3D
max value Originator Beneficiary
avg value Originator Beneficiary
avg value Intermediary Beneficiary
latency Intermediary Beneficiary
frequency with currency Intermediary Beneficiary
sum value Intermediary Beneficiary 3D
max value Intermediary Beneficiary 3D
frequency with currency Intermediary Beneficiary 3D

A. Features Extraction

We compute the features for the fields listed in Table VII.

Then, we compute the features related to transactions paths. Features are extracted for each transaction on a time window of one year, one month, and three days before the transaction date. We obtain 267 features, and we reduce this number using the *SelectFromModel* algorithm, a meta-transformer with a model trained with a classifier to assign an importance score to each feature. We use CatBoost as the classifier for its performance and capacity to deal with categorical features. The features number is an algorithm parameter, we choose 10 features based on SWIFT experts' knowledge (Table VIII).

The algorithm selects features related to the relationship between the intermediary and the beneficiary. For the transactions' history, no features related to the month were retained, except the last three days ('3D' suffix) and the last year. We apply a value transformation between 0 and 1 with the *Quantile Transformer* algorithm, which transforms according to a uniform distribution to reduce the outlier transaction impact. A very high amount of transactions could misrepresent other transactions during the visualization step (section IV-C).

B. Algorithms Comparison

We compare the classifiers from the literature to observe the best algorithm for our data. The results are presented in Table IX. Results show that CatBoost is the best algorithm for our data with a f1-score of 0.89.

[3] https://skaizengroup.eu/

[4] https://jupyter.org/

TABLE IX. CLASSIFIERS COMPARATIVE ACCORDING TO PRECISION, RECALL AND F1-SCORE

	Fraudulent			**Legitimate**			**All(average)**		
	Precision	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*
SVM	1.0	0.08	0.16	0.99	0.99	0.99	0.99	0.54	0.58
Random Forest	0.93	0.61	0.74	0.99	0.99	0.99	0.97	0.80	0.87
LightGBM	0.70	0.55	0.62	0.99	0.99	0.99	0.85	0.77	0.81
XGBoost	0.92	0.61	0.74	0.99	0.99	0.99	0.96	0.80	0.87
CatBoost	0.94	0.68	0.79	0.99	0.99	0.99	0.96	0.84	**0.89**

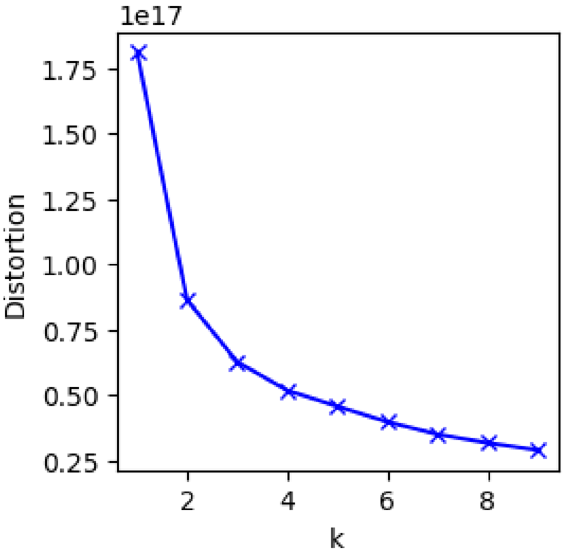


Fig. 2. Elbow Figure

TABLE X. CLUSTERS DISTRIBUTION

Cluster	Count of transactions	Average
0	4970	107999
1	4652	771069
2	3745	1279286

Our features are relevant enough to distinguish legitimate and fraudulent transactions. Our future experiment models will be trained with CatBoost.

C. Clustering of Fraudulent Transactions

After algorithms comparison, k-means [44] unsupervised algorithm is applied on the fraudulent transactions subset. We test different cluster numbers on our 13367 fraudulent transactions. We used the elbow method [45] to select the optimal k Fig. 2. The optimal cluster number is 3 because the curve linearly decreases at this number. Table X presents transactions distribution for each cluster and the average of transactions amount. The 3 clusters have an equivalent transactions number; however, the average of the transactions amount is different. The first cluster has a low average, the second one has a medium average, and the third has a high average.

In order to assign a fraud type to each cluster, we present fraud types in Table XI identified by our experts and the literature [46].

TABLE XI. FRAUD TYPES

Fraud Types	Description
Payment diversion	unauthorized redirection of payment instructions
Large amount	unauthorized initiation of high-value transactions
Smurf	illegal money laundering using money mules
Dormant account	unauthorized use of inactive bank accounts
False demand draft	creation and use of fake financial instruments to withdraw money.

D. Clusters Analyses and Fraud Types Identification

To analyze clusters, we used the SHAP framework [43] with the binary classification model trained on section IV-B on the transactions of each cluster. We compute the Shapley values with SHAP. For the interpretation, we use two visualization types: i) *beeswarm* for global cluster visualizations with features' importance and their value, ii) *heatmap* for cluster local visualizations with features impact on each transaction prediction.

1) *Cluster 0:* From the *heatmap* (Fig. 3), we split the map in two parts with the first five features : (i) on the left side, the first three features have a high impact. The *beeswarm* (Fig. 4) indicates that these values are low. For the last two features, their impact is lower, and their values are medium. (ii) The right side indicates the high impact of the first and fourth features with low values and a negative impact of the feature *value* (Amount field) with low values. This fraud type comprises customers making a few transactions with low and medium amounts.
Experts analyzed it as either to new customers doing few transactions (low frequency) of low amount (left side), or either to customers doing medium amount transaction after a long time (right side). This cluster is assigned to the "dormant account" fraud type (DAF).

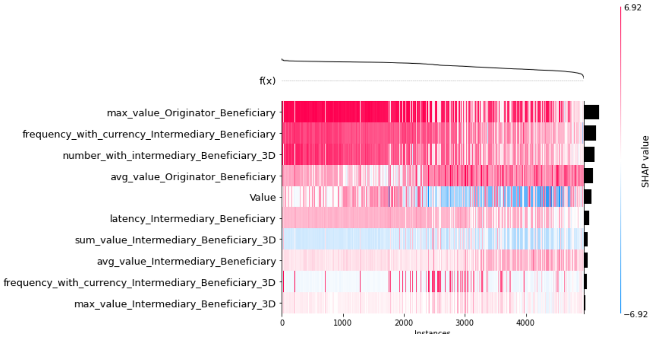


Fig. 3. Heatmap cluster 0

2) *Cluster 1:* According to the *heatmap* (Fig. 5), we split the map in two parts: (i) the right side shows that the amount

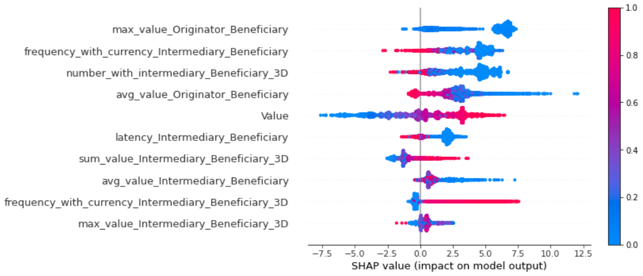


Fig. 4. Beeswarm cluster 0

has a high impact with high value, as shown in the *beeswarm* (Fig. 6). (ii) the *heatmap's* left side has a high impact on the second and third features with high values according to the *beeswarm*. The amount has an average impact corresponding to a medium value for these impacts.

Experts associated this fraud type with customers realizing many transactions (high frequency) in a short time (three days) and with an average amount. This cluster is assigned to the "smurf" fraud type (SF).

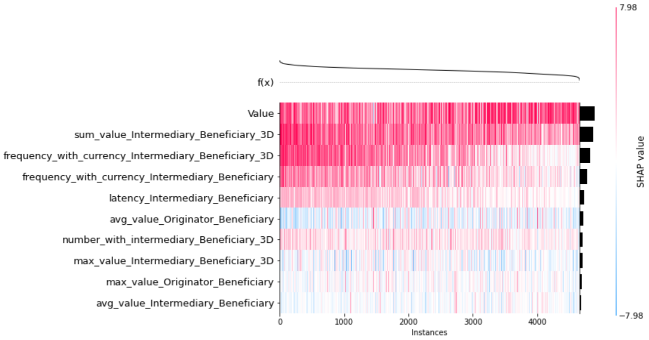


Fig. 5. Heatmap cluster 1

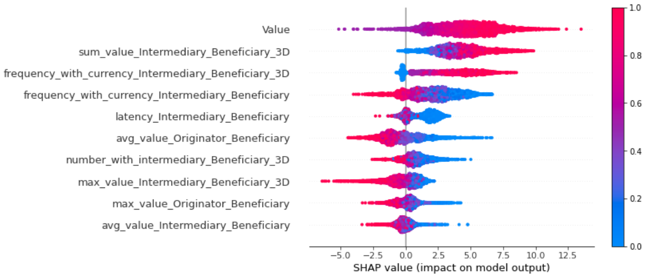


Fig. 6. Beeswarm cluster 1

3) Cluster 2: According to the *heatmap* (Fig. 7), the feature *value* (Amount field) greatly impacts the whole cluster. Furthermore, the *beeswarm* (Fig. 8) indicates that its value is high, and if we revert to Table X, the average amount of this cluster is very high. Experts associate this fraud type with customers realizing a high amount of transactions. This cluster is assigned to the "large amount" fraud type (LAF).

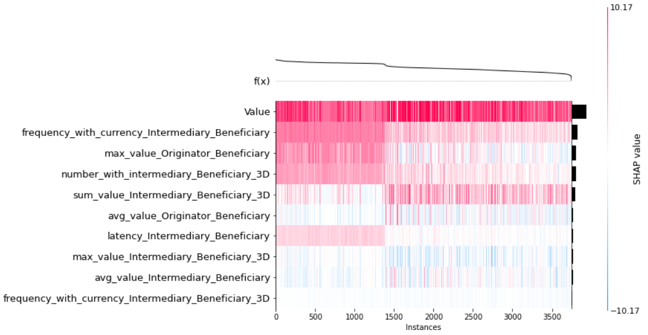


Fig. 7. Heatmap cluster 2

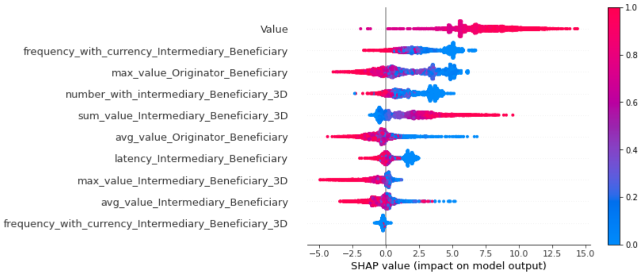


Fig. 8. Beeswarm cluster 2

We identified three fraud types by leveraging the binary classification model in combination with the SHAP framework and *k-means* algorithm. The next step is training a multi-fraud classification model to classify frauds in their fraud types. we update the fraudulent transaction label with their fraud type.

E. Multi-Fraud Classification and Mounting Cost Minimization

We train a multi-Fraud classification model with the same transactions in the training and testing set. Once our model is trained, we choose a threshold for the *MFD* algorithm presented in Section III-E.

we used our cost minimization algorithm to select the optimal threshold. The administrative cost C_a is set to 100, and the dissatisfaction cost C_d set to 50.

Fig. 9 shows the model mounting cost, and its f1-score is represented by two curves for each threshold. The mounting cost is low when the threshold is low because transactions probability to belong to fraudulent class are above 0.1. There is no cost impact on the model with fraudulent transaction amounts. However, the model generates many false alerts, for this reason, it's important to maintain a good f1-score. When the threshold is above 0.1, the model has the highest f1-score (0.85). Using our cost minimization algorithm, we retrieve 0.19 as the optimal threshold that minimizes the model mounting cost while maintaining a good f1-score (0.85). We reduce the f1-score of the hundredth order, which is insignificant.

We detail the model's evaluation results with the threshold in Table XII and in the confusion matrix in Table XIII. Our model has a good precision and a weaker recall. It results in a f1-score of 0.74, which is lower than 0.85 with the binary classification. *MFD* algorithm sums the probability of 3 fraud types. The transaction is fraudulent if the sum is above a threshold and its class is the fraud type with

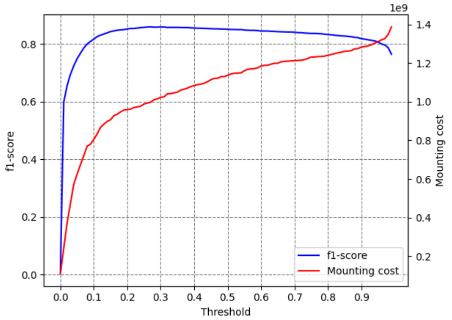


Fig. 9. Mounting costs and f1-score curves

TABLE XII. Multi-Class Model

	Fraud Type	Precision	Recall	f1-score
Multi-Class	Legit	0.99	0.99	0.99
	SF	0.91	0.43	0.58
	DAF	0.93	0.50	0.65
	LAF	0.92	0.60	0.73
	Macro avg	0.94	0.63	**0.74**
Binary (DAF+SF+LAF)	Legit	0.99	0.99	0.99
	Fraud	0.92	0.57	0.70
	Macro avg	0.96	0.78	**0.85**

the highest probability. In conclusion, our multi-class model detects fraudulent transactions as a binary model, and the fraud type assignation reduces the f1-score of just 0.09 but give additional information to experts about frauds.

V. Limits and Discussions

In this section, we completed the 3 goals outlined in section III through experimentation. First, we detect fraudulent transactions by extracting relevant features and using the *CatBoost* algorithm. This resulted in a f1-score of 0.89. However, due to data privacy limitations, we can't compare these results with other datasets. Second, we identify our dataset's fraud types based on the *CatBoost* model, *k-means* algorithm and SHAP framework. A multi-class classification model is trained using the *MFD* algorithm to detect fraudulent transactions and assign them a fraud type. Finally, we select a threshold to detect fraudulent transactions in order to achieve the highest f1-score while minimizing costs. Our multi-class model obtain 0.74 as f1-score (Table XII), which is lower than the 0.89 from the first step. However, if we consider a transaction as fraudulent when it's part on one the 3 fraud types, there is a f1-score of 0.85 (binary row). It means that fraudulent transactions are still detected on this model, however the fraud type assignation is decreasing the f1-score. It can be explained by possible close boundaries between clusters.

TABLE XIII. Confusion Matrix

		Predicted			
		Legitimate	DAF	SF	LAF
Actual	Legitimate	732605	18	2	89
	DAF	317	243	2	2
	SF	55	2	59	0
	LAF	779	2	0	1184

VI. Conclusion

We presented in this work the context of financial transactions between banks through the SWIFT network. We studied financial fraud detection literature. Machine learning techniques are valuable for identifying the fraudulent pattern provided during the training phase. We proposed a detection and identification frauds approach based on Desrousseaux et al. method. We can summarize our approach as: First, we extracted features from the transaction base fields and on the actors, currencies, countries, and transactions path. Second, we applied a supervised algorithm on our labeled dataset and reduced the features number to retain the relevant one. Third, we used the unsupervised algorithm to cluster fraudulent transactions to identify the fraud types by leveraging the SHAP framework and the supervised model. Then, we trained a multi-fraud classification model to assign a class to a transaction with the three identified fraud types: dormant account fraud, smurf fraud, and large amount fraud. To handle this model prediction, we proposed the *MFD* algorithm to classify a transaction as fraudulent or fraudulent with its type. Finally, we proposed another algorithm to minimize the model mounting cost by choosing a threshold from which a transaction is considered fraudulent.

Our semi-supervised methodology is used by financial institutions to understand their dataset. Cluster interpretation needs experts feedback based on visualization tools (Fig. 3-8).

In conclusion, our contributions with the proposed approach are : detecting fraudulent transactions, identifying fraud types and minimizing the mounting cost.

In future work, we plan to explore additional fraud types on different datasets. We also plan to automatize identifying fraud type with semantic analyses based on financial fraud ontology.

Acknowledgment

The authors would like to thank the University of Burgundy and more particularly the laboratory LIB for the scientific help. We also thank SKAIZen Group company for providing the dataset and experts. Finally, we thank the ANRT for supporting this project.

References

[1] D. Jensen, "Prospective assessment of ai technologies for fraud detection: A case study," in *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*. Citeseer, 1997, pp. 34–38.

[2] S. Dasgupta and P. Grover, "Critically evaluating swift's strategy as a monopoly in the fintech business," 2019.

[3] M. Collin, S. Cook, and K. Soramaki, "The impact of anti-money laundering regulation on payment flows: Evidence from swift data," *Center for Global Development Working Paper*, no. 445, 2016.

[4] J. L. Perols, "Detecting financial statement fraud: Three essays on fraud predictors, multi-classifier combination and fraud detection using data mining," 2008.

[5] M. Dixon, I. Halperin, and P. Bilokon, *Machine Learning in Finance: From Theory to Practice*. Springer International Publishing, 2020. [Online]. Available: https://books.google.fr/books?id=Fuw3zQEACAAJ

[6] A. Kumar, S. Das, and V. Tyagi, "Anti money laundering detection using naïve bayes classifier," in *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, 2020, pp. 568–572.

[7] U. Porwal and S. Mukund, "Credit card fraud detection in e-commerce: An outlier detection approach," *arXiv preprint arXiv:1811.02196*, 2018.

[8] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information sciences*, vol. 557, pp. 317–331, 2021.

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

[10] L. Keyan and Y. Tingting, "An improved support-vector network model for anti-money laundering," in *2011 Fifth International Conference on Management of e-Commerce and e-Government*. IEEE, 2011, pp. 193–196.

[11] X. Wang and G. Dong, "Research on money laundering detection based on improved minimum spanning tree clustering and its application," in *2009 Second international symposium on knowledge acquisition and modeling*, vol. 2. IEEE, 2009, pp. 62–64.

[12] R. A. L. Torres and M. Ladeira, "A proposal for online analysis and identification of fraudulent financial transactions," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 240–245.

[13] D. Almhaithawi, A. Jafar, and M. Aljnidi, "Example-dependent cost-sensitive credit cards fraud detection using smote and bayes minimum risk," *SN Applied Sciences*, vol. 2, no. 9, pp. 1–12, 2020.

[14] E. A. Lopez-Rojas and S. Axelsson, "Money laundering detection using synthetic data," in *Annual workshop of the Swedish Artificial Intelligence Society (SAIS)*. Linköping University Electronic Press, Linköpings universitet, 2012.

[15] J. Tang and J. Yin, "Developing an intelligent data discriminating system of anti-money laundering based on svm," in *2005 International conference on machine learning and cybernetics*, vol. 6. IEEE, 2005, pp. 3453–3457.

[16] K. Michalak and J. Korczak, "Graph mining approach to suspicious transaction detection," in *2011 Federated conference on computer science and information systems (FedCSIS)*. IEEE, 2011, pp. 69–75.

[17] L.-T. Lv, N. Ji, and J.-L. Zhang, "A rbf neural network model for anti-money laundering," in *2008 International conference on wavelet analysis and pattern recognition*, vol. 1. IEEE, 2008, pp. 209–215.

[18] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection-machine learning methods," in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, 2019, pp. 1–5.

[19] B. Bestami Yuksel, S. Bahtiyar, and A. Yilmazer, "Credit card fraud detection with nca dimensionality reduction," in *13th International Conference on Security of Information and Networks*, 2020, pp. 1–7.

[20] M. Alkhalili, M. H. Qutqut, and F. Almasalha, "Investigation of applying machine learning for watch-list filtering in anti-money laundering," *IEEE Access*, vol. 9, pp. 18 481–18 496, 2021.

[21] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision support systems*, vol. 50, no. 3, pp. 602–613, 2011.

[22] E. L. Paula, M. Ladeira, R. N. Carvalho, and T. Marzagao, "Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering," in *2016 15th ieee international conference on machine learning and applications (icmla)*. IEEE, 2016, pp. 954–960.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[24] E. Badal-Valero, J. A. Alvarez-Jareño, and J. M. Pavía, "Combining benford's law and machine learning to detect money laundering. an actual spanish court case," *Forensic science international*, vol. 282, pp. 24–34, 2018.

[25] N. F. Ryman-Tubb, P. Krause, and W. Garn, "How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark," *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 130–157, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197618301520

[26] K. G. Al-Hashedi and P. Magalingam, "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019," *Computer Science Review*, vol. 40, p. 100402, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013721000423

[27] M. Albashrawi, "Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015," *Journal of Data Science*, vol. 14, pp. 553–570, 07 2016.

[28] Y. Zhang and P. Trubey, "Machine learning and sampling scheme: An empirical study of money laundering detection," *Computational Economics*, vol. 54, no. 3, pp. 1043–1063, 2019.

[29] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, "Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–8.

[30] A. Mehbodniya, I. Alam, S. Pande, R. Neware, K. P. Rane, M. Shabaz, and M. V. Madhavan, "Financial fraud detection in healthcare using machine learning and deep learning techniques," *Security and Communication Networks*, vol. 2021, 2021.

[31] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[33] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[34] A. S. Larik and S. Haider, "Clustering based anomalous transaction reporting," *Procedia Computer Science*, vol. 3, pp. 606–610, 2011.

[35] J. Guo, G. Liu, Y. Zuo, and J. Wu, "An anomaly detection framework based on autoencoder and nearest neighbor," in *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*. IEEE, 2018, pp. 1–6.

[36] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[37] N. A. Le Khac and M.-T. Kechadi, "Application of data mining for anti-money laundering detection: A case study," in *2010 IEEE International Conference on Data Mining Workshops*. IEEE, 2010, pp. 577–584.

[38] S. Raza and S. Haider, "Suspicious activity reporting using dynamic bayesian networks," *Procedia Computer Science*, vol. 3, pp. 987–991, 2011.

[39] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decision Support Systems*, vol. 133, p. 113303, 2020.

[40] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost sensitive credit card fraud detection using bayes minimum risk," in *2013 12th international conference on machine learning and applications*, vol. 1. IEEE, 2013, pp. 333–338.

[41] R. Desrousseaux, G. Bernard, and J.-J. Mariage, "Profiling money laundering with neural networks: a case study on environmental crime detection," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2021, pp. 364–369.

[42] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[43] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[44] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[45] T. M. Kodinariya and P. R. Makwana, "Review on determining number of cluster in k-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[46] A. D. Boyer and S. Light, "Dirty money and bad luck: Money laundering in the brokerage context," *Va. L. & Bus. Rev.*, vol. 3, p. 81, 2008.