IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 13, 2020, accepted August 26, 2020, date of publication September 7, 2020, date of current version September 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022315

A Closer Look Into the Characteristics of Fraudulent Card Transactions

BARIS CAN[1], **ALI GOKHAN YAVUZ**[2], **ELIF M. KARSLIGIL**[2], **AND M. AMAC GUVENSAN**[2], **(Member, IEEE)**
[1]Scientific and Technological Research Council of Turkey (TUBITAK), 41470 Istanbul, Turkey
[2]Department of Computer Engineering, Yildiz Technical University, 34220 Istanbul, Turkey

Corresponding author: M. Amac Guvensan (amac@yildiz.edu.tr)

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 3170946.

ABSTRACT Widespread use of Internet also had the substantial impact on the increase of the online card transactions especially with the beginning of the last decade. Along with the increase of online transactions, the worldwide banking sector was forced to deal with or to encounter an unforeseen number of fraudulent activities, yet. Hence, rule-based systems were designed to mark the high-risk transactions and let the experts to confirm the fraudulent nature of such transactions. As a countermeasure, static nature of rule-based systems were exploited by the latest attacks to go undetected. Thus, researchers aimed at designing adaptive fraud detection systems utilizing mainly machine learning techniques with the very recent application of deep learning. However, they were focused on detecting fraudulent activities but, to the best of our knowledge, none of them delved into the better understanding the characteristics of fraudulent card transactions in order to produce more resilient models. Therefore, in this study, we built the biggest data set ever used in a research, consisting of 4B non-fraud and 245K fraud transactions contributed to by the 35 banks in Turkey. Consequently, we introduce and examine the performance of profile-based fraud detection models, namely card-type based model, transaction characteristics based model, and amount-based model. Also, we made temporal and spatial analysis on our data set to show the robustness of the proposed models against aging and zero-day attacks.

INDEX TERMS Fraud detection, profiling, amount range, card-type, transaction characteristics, zero-day attack, amount-based success rate.

I. INTRODUCTION

The number of credit card transactions are increasing in line with technological developments and the rise of e-commerce. In 2017 alone, 375 billion card payments were made around the world [1]. However, 16.7 million fraudulent transactions occurred in the same year [2]. The ratio of fraudulent transactions to normal transactions is approximately 0,006% worldwide. Although this rate may seem insignificant, every fraudulent transaction hurts the reputation of banks. For this reason, banks are investing in fraud detection. The number of fraudulent activities and their methods increases and changes every day. It is very difficult and costly to detect fraudulent activities only by examining the transactions. Fast and accurate fraud detection is crucial to maintain customer satisfaction and trust. Therefore, banks need to identify these transactions as quickly as possible and in the least harmful way for the customer.

Today, fraudulent activities using social engineering are predominantly performed through Internet. Malware and phishing methods are engineered for this purpose. Most popular types of fraud include customer information altering through call center and branches, ATM fraud, credit card application fraud, card account theft, lost-stolen, fake credit cards and card duplication [3].

Current commercial solutions used by banks for fraud detection are primarily rule-based. However, in recent years research has shown that machine learning methods are more effective than most rule-based solutions. The data sets used in the publications in which these results are published do not always correspond to the real banking environment in terms of numbers, characteristics, and changes in time. In this paper, unprecedented analysis on a real data set were conducted to reveal the unidentified characteristics of fraud detection activities. 245,000 fraudulent transactions and four billion

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqiang Wang.

non-fraudulent transactions obtained from different banks for the year 2017 were utilized in these analyses.

This study makes the following contributions to the literature:

- To the best of our knowledge, a data set with the largest amount of fraudulent transactions was created. Thus, all the analysis were carried out and experimental results were obtained using this data set.
- Card transactions were profiled based on card-type, amount and transactional characteristics. The resulting models were shown to have negligible effect on the fraud detection performance when compared to similar models applied to the unprofiled data set.
- The performance of fraud detection models using transactional characteristics were shown to decay with time. Thus, such models require periodical training with recent fraud and non-fraud instances.
- We attempted to evaluate the zero-day performance of the models using both unprofiled and profiled data sets. We observed the behaviour of the models against the unseen fraudulent transactions.
- In contrast to existing studies on cost-sensitive fraud detection models, we aimed at expressing the performance of fraud detection models in terms of financial gain and loss.

The manuscript is outlined as follows. Section II discusses state-of-the-art studies and points out our novel contributions to the literature. In Section III, we introduce BKM data set especially how it satisfies big data characteristics and its importance to the fraud detection research domain. Section IV elaborates on the details of pre-processing and feature selection applied to BKM data set, and we also compare fraud detection models in terms of their eligibility to perform the anticipated analysis tasks on the data set. In Section V, both the outlines and the results of the analysis are given in depth. Then, we conclude the paper in Section VI.

II. RELEATED WORK

Fraud detection is very popular and is practiced in multiple areas. The survey of Abdallah *et al.* examined a wide range of fraud topics including [4] credit card fraud [5]–[7], telecommunication fraud [8], [9], healtcare insurance fraud [10]–[12], automobile insurance fraud [13], [14] and online auction fraud [15], [16]. The survey revealed that the majority of the previous researches were conducted on banking fraud. Banking fraud is followed by insurance fraud, e-commerce fraud, and telecommunications fraud. The size and uneven distribution of the data is one of the biggest problems with fraudulent banking transactions. Many studies have focused on solving this problem. However, in order to solve the given problems, many studies establish fraud detection models by randomly selecting a certain number of transactions among non-fraudulent transactions [17]–[20].

The privacy and protection rules of personal data have further complicated access to real banking transactions for research purposes. Ong Shu Yee *et al.* mimicked real life data to overcome this problem [21]. In this study, particular attributes, such as credit card number, reference number and terminal id were determined and mimicked to create synthetic transactions. In the study conducted by Andrea Dal Pozzolo *et al.*, unlike other studies, a real system was designed and co-utilization of data-driven and rule-based methods were suggested, along with periodic updates [22].

Many fraud detection studies focus on supervised methods, such as Naive Bayes [19], [23], Random Forest [19], [20], [23], Support Vector Machine (SVM) [23], [24], Artificial Neural Networks (ANNs) [17], [19], [23], Deep Learning [25], Bayesian Belief Network [17], k-Nearest Neighboor (k-NN) [19], [23], Decision Tree such as C4.5 [20], Logistic Regression [18], [20]. The highest success in fraud detection was generally achieved by Bayesian Network classifiers, and the highest success with non-fraud detection was achieved by Random Forest and Decision Tree approaches [19], [21], [23]. On the other hand, there are alternative solutions [26], [27] including undersampling technique [28], hybrid approach [29], behavioral/historical analysis of transactions [30], [31] with promising results and few deep learning approaches applied on credit card fraud detection problem [32].

Among the reviewed studies, some of them were found to be comparable to ours in terms of methods, test scenarios, and performance metrics. Shiyang Xuan *et al.* created various classification scenarios and used the B2C transactions obtained from a Chinese e-commerce site between November 2016 and January 2017 [33]. They worked with a data set containing more than 30 million transactions with 62 features. However, their data set contained only 82,000 fraudulent transactions. In this study, Random-Tree Based and Classification and Regression (CART)-Based Random Forest algorithms were compared. The best accuracy (96.77%) and the best F-measure value (0.9691) were obtained by the CART-Based Random Forest algorithm. Training and test groups were formed and tested with the rates 1:1 to 10:1. This was to identify the significance of the non-fraud:fraud ratio. Only January transactions were included in the data sets. The results show a continuous increase in accuracy. This was attributed to the enlargement of the test data and training sets. In order to assure fair comparisons, the data set must be kept constant. Recall rate tends to decrease for fraudulent transactions. The best F-measure value is 0.964 at the 5:1 ratio. Finally, the first 11 days of January 2017 were tested against the 2016 data in training. For fraudulent transactions, the recall rate was 59.62% and the accuracy of the test was 98.67%. The recall rate shows that when the test is performed with data from months excluded from training, fraud detection success decreases significantly.

Study [21] established the importance of the pre-transaction stage and unlike other studies, also included the test time as a performance metric. In this study, normalization, smoothing, aggregation, attributes construction, and generalization of the data were carried out during the pre-transaction stage, then dimensionality reduction was performed using the Principal Component Analysis (PCA) technique. The experimental results in this study were obtained using 10-fold cross validation. In the study, Bayesian classifiers such as K2, Tree Augmented Naive Bayes (TAN) and Naive Bayes were used along with Logistic Regression and J48 (C4.5) algorithms. The highest accuracy for raw data belonged to the TAN algorithm at 84.0%, while the worst success rate belonged to the K2 algorithm with 41.8%. The success rates increased when engineered features were used. According to their results, the Logistic Regression and J48 decision tree algorithm scored 100% accuracy. The lowest accuracy rate was 95.8% and with the K2 algorithm. However, these success rates were achieved using a synthetic data set. It is hard to tell whether the model will be as successful in real life situations. As to the execution times, TAN and J48 algorithms were found to be taking longer to execute than others, while K2 and Naive Bayes were fastest to produce results.

Some studies preferred adopting ensemble approach, such as AdaBoost and Majority Voting. In study [34], a publicly available data set [28] containing 284,807 transactions (492 fraudulent) made in September 2013 by European card-holders was used to generate experimental results. The results were obtained by using the 10-fold cross validation method. The best fraud detection success belonged to Naive Bayes with 83.13% and the best non-fraud detection success belonged to the Random Forest algorithm with 99.99%. The Adaboost and Majority Voting methods have decreased fraud detection success and increased non-fraud detection success. This increase in accuracy was attributed to the data set containing more non-fraud data. There are 287,224 transactions in the data set created by the researchers. Of these, 102 are marked as fraudulent. In tests conducted with this data set, initial fraud detection success was over 90% and non-fraud detection success was 99.99% for various algorithms. The Adaboost and Majority Voting methods were observed to increase fraud detection success. However, the number of fraudulent transactions in both data sets are insufficient to conclude on the success of the models.

The study published by Abhimanyu Roy *et al.* in 2018 aimed at observing the effect of deep learning methods on credit card fraud detection and used the data set provided by financial institutions engaged in retail banking [25]. The data set contains about 80 million transactions collected over an eight-month period. Only 0.14% of the data is fraudulent. They used four different Deep Learning topologies: Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTMs), and Gated Recurrent Units (GRUs). In the study, classification results were obtained by using 10-fold cross validation. The best accuracy of the study was achieved by the GRU topology with 91.6%. The study claimed that increasing the number of layers and nodes led to higher success rates.

Some studies have used unsupervised methods [35] to detect fraud. Richard J. Bolton *et al.* stated that it would not always be possible to access fraudulent transactions, meaning that the success achieved in the supervised methods was misleading [36]. This study calculated a suspicion score for each user. This score was updated with every transaction. The Peer Group Analysis (PGA) tool was developed to observe the change in spending behavior. The study involved weekly analysis of total expenditures for 858 accounts in 52 weeks. The PGA was applied to four-week periods for each account.

TABLE 1. An overview of the fraud data sets in the literature is given below. Their characteristics are compared against our data set.

Dataset	Non-fraud	Fraud	Total
E-commerce [33]	30M	82k	30M
European Cardholderster [38]	284k	492	284k
Kuldeep Randhawa [34]	287k	102	287k
Abhimanyu Roy [25]	80M	112k	80M
BKM (Our Dataset)	4,000M	245k	4,000M

Apapan Pmsirirat *et al.* have argued that methods of fraud are constantly changing, and therefore unsupervised methods should be used [37]. 80% of the data sets were used for training, while 20% were spared for testing. Every instance had 21 features. In their study, anomaly detection in customer profiles was carried out using the Auto Encoder and Restricted Boltzmann machine (RBM) methods. In the Auto Encoder based system, Area Under Curve (AUC) values was 0.9603 for the European data set. The sizes of the data sets referenced in this section are presented in Table 1.

In contrast to the data sets mentioned in the literature, our data set is composed of purely real financial transactions of debit and credit cards issued by 35 different banks in Turkey. The data set spans a time period of 8 months and is comprised of more than 4 billion transactions of which 0.006% was reported as fraudulent activity by the respective banks. Apart from other studies, we investigated the effect of profile-based fraud detection models on the performance. As the basis of the profile-based fraud detection, we clustered the data set according to *card-type, amount spent, and transactional characteristics*. Then, the transactions were tested for fraudulent activity using the respective model. In addition to the classical temporal performance decay analysis of the fraud detection models, we also formulated the zero-day attack performance to this end. We specifically left out some of the clusters generated as a result of k-means clustering of the fraudulent instances from the training sets. Those clusters were included in the test data sets to observe the performance of the respective model in case of a never-before seen fraudulent activity. As in medical diagnosis problems or network intrusion detection scenarios, our data set is imbalanced by nature. Although current literature includes cost-sensitive models [39], [40] [41] especially where data imbalance problem becomes prominent, we attempted to measure the performances of the fraud detection models in terms of financial value.

III. DATASET

In this study, we use real banking data obtained from the banking sector in Turkey. The data set contains more than

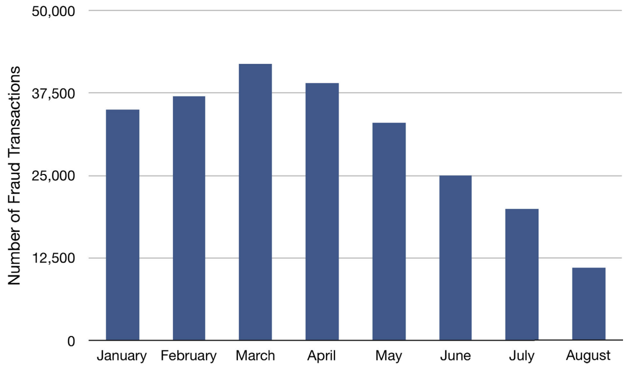


FIGURE 1. The monthly distribution of fraudulent transactions, 2017.

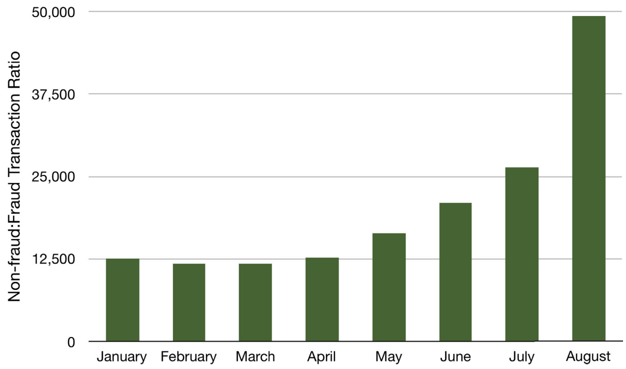


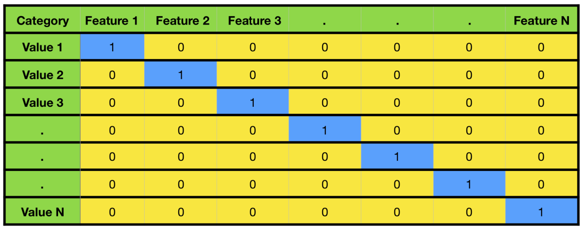
FIGURE 2. The monthly ratio of non-fraudulent transactions to fraudulent transactions, 2017.

TABLE 2. The characteristics of the features in our data set.

	Number of	
Feature Categories	**Features**	**Viable Features**
Transaction	17	13
Card	14	9
Merchant	9	5
Bank	8	5
POS	8	5
Time	4	1
Total	60	38

four billion credit and debit card transactions belonging to 35 member banks between January 2017 and August 2017. The ratio of fraudulent transactions to non-fraudulent ones is 0.006%.

The number of fraudulent transactions per month are given in Figure 1, whereas Figure 2 depicts the ratios of non-fraud to fraud transactions. Table 3 gives the detailed distribution of transactions regarding their card type, amount range, and transaction type. Each transaction in the data set contains 60 features. It was observed that many features were mis-valued and did not have an even distribution. Twenty-two such features were eliminated in the first stage. Of the remaining 38 viable features, three are numerical and the rest are categorical. The types and counts of the initial and viable features are given in Table 2.



Category	Feature 1	Feature 2	Feature 3	.	.	.	Feature N
Value 1	1	0	0	0	0	0	0
Value 2	0	1	0	0	0	0	0
Value 3	0	0	1	0	0	0	0
.	0	0	0	1	0	0	0
.	0	0	0	0	1	0	0
.	0	0	0	0	0	1	0
Value N	0	0	0	0	0	0	1

FIGURE 3. Conversion of a categorical feature consisting of N distinct values into the corresponding One-Hot Vector with N features.

IV. FRAUD DETECTION MODEL

In order to create a fraud detection model, incomplete or incorrect data was eliminated, distinguishing features were identified, the instances to be used in the model were selected, and the performance of classification algorithms were evaluated.

Analyses were performed on the whole data set. The data set were split up by 70% to 30% for training and testing, respectively. The training part of the data set was further divided into two sub data sets with a ratio of 70% to 30% for training and validation, respectively. This data set will be referred to as the master data set throughout the remaining of this study. Thus, master training data set should be interpreted as the training part of the master data set as described above.

A. PRE-PROCESSING

As the majority of the viable features were categorical, most of the pre-processing efforts were put into dealing with them. For each categorical feature category values were analyzed and invalid ones were replaced with null values. To represent categorical values, the one-hot encoding method was used, being one of the most common methods for converting categorical features to numerical ones.

The one-hot vector is obtained by expressing each distinct value for a feature in binary form. As a result, the number of features increases by the number of existing distinct values for each feature. In every one-hot vector, the feature belonging to the respective category is expressed as "1", and other features are expressed as "0". This process is shown in Figure 3.

Numerical features were normalized into the [0,1] range using min-max normalization technique [42].

B. FEATURE SELECTION

On a dataset, comprised of all fraudulent transactions plus randomly chosen non-fraudulent transactions with an equal number (Table 4), *Information Gain [43], Gain Ratio [44], One Rule [45], Relief [46], Symmetrical Uncertainty [47]* algorithms were used for feature selection, and ranker values were obtained. The first 5, 10, 15, 20, 25, 30, 35, 38 features were grouped according to their ranker value and classification tests were conducted to decide the features that were going to be used. The accuracy values given in Figure 4 were evaluated and the first 30 common features were chosen.

TABLE 3. The detailed distribution of transactions regarding their card type, amount range and their transaction type.

	Jan.		Feb.		Mar.		Apr.		May		Jun.		Jul.		Aug.	
Card Type	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**
Business(%)	12.21	11.31	9.59	10.99	8.86	11.31	10.87	11.08	8.38	11.31	9.03	11.08	10.13	11.31	13.19	11.31
Classic(%)	46.50	24.24	47.39	23.95	45.01	24.24	43.42	24.17	40.65	24.24	44.07	24.17	41.03	24.24	50.38	24.24
Debit(%)	8.22	43.95	4.88	44.96	5.67	43.95	8.37	44.38	11.47	43.95	10.17	44.38	11.33	43.95	7.15	43.95
Gold(%)	33.07	20.50	38.14	20.10	40.46	20.50	37.34	20.36	39.50	20.50	36.73	20.36	37.51	20.50	29.27	20.50
Amount Range	**Jan.**		**Feb.**		**Mar.**		**Apr.**		**May**		**Jun.**		**Jul.**		**Aug.**	
(Turkish Lira)	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**	**NF**	**F**
$\mathbf{0-1^*}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\mathbf{1-10}$	0.03	0.27	0.03	0.27	0.02	0.29	0.03	0.28	0.02	0.28	0.02	0.23	0.03	0.26	0.02	0.23
$\mathbf{10^1\text{-}10^2}$	14.45	10.67	16.46	10.42	13.55	10.43	15.14	10.33	13.24	10.31	11.17	9.99	8.20	10.11	7.04	9.37
$\mathbf{10^2\text{-}10^3}$	36.34	35.47	41.72	34.94	40.81	34.83	43.11	35.17	43.11	35.28	43.48	35.34	46.60	35.61	43.49	33.53
$\mathbf{10^3\text{-}10^4}$	41.94	43.58	38.82	44.75	38.07	45.04	31.20	46.03	37.31	45.08	36.90	46.12	37.21	45.04	42.80	48.48
$\mathbf{10^4\text{-}10^5}$	7.24	7.95	2.98	7.70	7.55	8.37	8.80	7.38	6.32	8.17	8.44	7.46	7.97	8.04	6.63	7.59
$\mathbf{10^5\text{-}10^6}$	0.00	1.97	0.00	1.82	0.00	0.87	1.74	0.72	0.00	0.82	0.00	0.77	0.00	0.85	0.00	0.73
$\mathbf{>10^6}$	0.00	0.09	0.00	0.10	0.00	0.17	0.00	0.09	0.00	0.08	0.00	0.09	0.00	0.08	0.00	0.06

*:The percentages look as 0.00 due to too small values

TABLE 4. The number of instances in train and test sets are given below. All fraudulent transactions and randomly selected non-fraudulent transactions regarding 1:1 ratio are split into train (70%) and test sets (30%).

Dataset	Non-fraud	Fraud
Train	171k	171k
Test	73k	73k

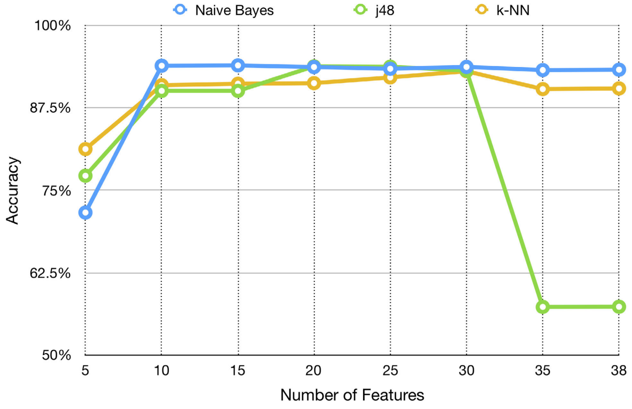


FIGURE 4. The relationship between accuracy and number of features.

As a result of the feature selection process, a vector with a length of 4,177 was obtained from the selected 30 features of which 29 categorical ones are represented by a 4,176 long one-hot vector and the remaining one by real values in the [0,1] range. The types and numbers of selected features are given in Table 5. Table 6 reflects the detailed analysis of the selected features and their categories in terms of fraudulent distribution.

C. INSTANCE SELECTION

Since the fraud:non-fraud ratio is about 0.01%, instance selection was carried out to establish a balance within the data set. Different methods were adopted to select non-fraudulent transactions, while all of the fraudulent transactions were used in the analysis.

TABLE 5. The distribution of selected features according to their characteristics.

Feature Type	Number of Selected Features
Transaction	12
Card	8
Merchant	3
Bank	3
POS	3
Time	1
Total	30

Many studies in the literature suggested random selection of non-fraudulent transactions [17]–[20], [33]. Therefore, non-fraudulent transactions were randomly selected in this study as well. However, to better reflect the characteristics of the transactions to the features, the transactions were first clustered using the k-Prototypes algorithm [48] and the optimum number of groups were determined by using the elbow technique [49]. Non-fraudulent transactions were then randomly selected, not from the overall data set, but from each group. As pointed out in the literature mentioned above, the ratio of non-fraudulent to fraudulent transactions in the training group affects the performance of the model. Therefore, training data sets were formed using the 1:1, 5:1, 10:1, 15:1, 20:1, 25:1, 30:1, 35:1, 40:1 ratios, respectively, and classification tests were performed to decide the optimum ratio of non-fraudulent to fraudulent transactions. Non-fraudulent transactions were selected incrementally. Following the tests conducted using classification algorithms in Section IV-D, best f-measure value was found to be 5:1 ratio, which is the same conclusion reached in the study [33]. Consequently, we decided to use the 5:1 ratio in the remainder of our study.

D. CLASSIFICATION

As suggested in [17], [19], [20], [23], we considered to use the following machine learning algorithms for the classification process; Naive Bayes, Decision Tree [50], Random Forest [51] and Multi-Layer Perceptron [50]. The Naive Bayes [50] algorithm performs the classification process by calculating probabilities. For Naive Bayes classification,

TABLE 6. The characteristics of the features in our data set.

			Non-Fraud(%)					Fraud(%)				
Feature Category	**Feature Name**	**#categories**	**#1**	**#2**	**#3**	**#4**	**#5**	**#1**	**#2**	**#3**	**#4**	**#5**
Bank	Feature 1	2	93.84	6.16	-	-	-	92.28	7.72	-	-	-
Bank	Feature 2	62	20.70	16.06	15.25	5.73	10.99	43.31	29.52	14.13	8.17	0.57
Bank	Feature 3	30	45.45	8.34	11.96	8.46	7.22	58.39	21.52	16.18	2.75	0.05
Card	Feature 4	11	56.29	30.23	12.31	1.06	0.09	88.82	9.84	0.30	1.01	0.03
Card	Feature 5	2	47.60	52.40	-	-	-	98.47	1.52	-	-	-
Card	Feature 6	62	13.29	17.90	13.78	14.98	10.14	32.13	16.20	9.01	4.28	7.38
Card	Feature 7	3	57.57	41.44	0.95	0.03	-	96.90	3.10	0.00	0.00	-
Card	Feature 8	8	48.69	50.74	0.44	0.07	0.04	60.61	39.39	0.00	0.00	0.00
Card	Feature 9	31	50.96	2.08	7.60	33.69	4.09	0.93	36.96	30.70	1.83	28.95
Card	Feature 10	7	33.76	21.20	19.24	13.71	8.27	62.18	29.70	1.16	1.33	0.59
Merchant	Feature 11	2864	2.09	1.27	21.72	13.94	8.37	44.13	26.27	0.35	1.52	0.68
Merchant	Feature 12	277	98.10	0.54	0.36	0.26	0.14	100.00	0.00	0.00	0.00	0.00
Merchant	Feature 13	27	3.36	23.95	23.69	4.33	10.43	70.40	1.76	1.57	10.94	0.72
Merchant	Feature 14	84987	66.55	20.13	10.06	0.14	1.12	56.97	22.79	6.14	3.46	0.69
POS	Feature 15	2	65.36	17.79	16.86	-	-	40.70	58.76	0.54	-	-
POS	Feature 16	18	40.67	39.19	14.39	4.91	0.50	48.66	23.88	16.33	10.93	0.17
POS	Feature 17	9	65.93	14.67	2.96	1.79	8.40	3.64	43.79	32.67	19.46	0.24
Time	Feature 18	31	3.48	3.37	3.65	3.42	3.37	4.39	4.46	3.87	3.93	3.81
Transaction	Feature 19	2	64.73	33.09	2.18	-	-	74.88	24.89	0.23	-	-
Transaction	Feature 20	2	86.06	13.94	-	-	-	98.48	1.52	-	-	-
Transaction	Feature 21	34	51.19	47.73	0.77	0.28	0.02	98.98	0.96	0.06	0.00	0.00
Transaction	Feature 22	2	97.28	2.71	-	-	-	100.00	0.00	-	-	-
Transaction	Feature 23	15	89.10	3.52	1.03	3.31	2.29	4.21	62.20	29.49	2.59	0.73
Transaction	Feature 24	3	95.85	3.90	0.24	-	-	98.45	1.55	0.00	-	-
Transaction	Feature 25	29	52.36	14.02	7.12	7.74	9.26	22.96	55.94	15.91	2.75	1.05
Transaction	Feature 26	30	79.26	20.18	0.51	0.03	0.01	98.07	1.56	0.14	0.23	0.00
Transaction	Feature 27	2	99.92	0.00	0.08	-	-	96.03	3.97	0.00	-	-
Transaction	Feature 28	644	42.13	44.27	6.55	6.47	0.01	58.12	37.86	0.03	0.02	2.32
Transaction	Feature 29*	-	-	-	-	-	-	-	-	-	-	-
Transaction	Feature 30	72	88.88	7.22	0.98	0.69	0.62	98.12	0.33	0.44	0.30	0.16

*:Feature 29 consists of continuous values.

Bernoulli Naive Bayes [52] classifier was used as in the pre-processing step, the features were converted into one-hot vectors.

The classification results in terms of recall, specificity, precision, f-measure and mcc (matthews correlation coefficient) are given in Table 7. Based on recall values for the fraud class, Naive Bayes classifier outperforms the other classifiers, whereas the other classifiers were more performant for the classification of non-fraud instances. A closer look at Table 7 reveals the fact that the Naive Bayes classifier produces a rather large number of false alarms compared to the other classifiers. Our cursory tests show that for a randomly selected month, the ratio for the number of false alarms lies in the range of 6:1 in favor of other classifiers. From a financial point of view, this fact could be considered as a major drawback for the Naive Bayes classifier as it means more operational work for a financial institution to put the results of the classifier into action in real life scenarios.

V. ANALYSIS OF FRAUDULENT TRANSACTIONS

In this section, the data set was assessed spatially and temporally, the success of the system against zero-day fraud was analyzed, the financial gain from detecting non-fraudulent and fraudulent transactions was examined, and finally, the successes and run times of the algorithms were compared. Scikit-learn library was utilized to implement the classification processes used in the analyses [53]. The performance metrics provided in Equation 1, 2, 3, 4 were used in the study.

$$Precision = \frac{CorrectlyClassifiedFraudInstances}{ClassifiedasFraudInstances} \tag{1}$$

$$Recall = \frac{CorrectlyClassifiedFraudInstances}{FraudInstances} \tag{2}$$

$$F-Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

$$Specificity = \frac{CorrectlyClassifiedNonFraudInstances}{NonFraudInstances} \tag{4}$$

Initial investigation of the performances of the selected classifiers were carried out on the master data set as a whole. However, due to the random selection nature of both non-fraudulent and fraudulent instances for training, validation, and testing steps, for a given point in time the respective data sets could contain instances from the future. Thus, in order not to make the model to learn about fraud types not occurred yet and to produce better recall values which could not be obtained in real life, we opted for not including any instance within the training and validation data sets to obtain a better judgment about the performance of a given classifier. The test instances were naturally selected from transactions in future which reflects real life behaviour. Table 8 reflects the results of this approach. Starting with January 2017, one month's instances were used to train the respective model and the following month's instances were used to test the performance of the model. Again, during training the used

TABLE 7. The recall, specificity, precision and MCC values with regard the classes for the selected classifiers using the training and test data sets given in Section III.

Algorithms	NF	NF	NF	F	F	F	
	Precision	Specificity	F-Measure	Precision	Recall	F-Measure	MCC
Naive Bayes	99.24%	92.90%	95.96%	73.20%	96.44%	83.22%	80.44%
Random Forest	98.54%	98.97%	98.75%	94.77%	92.71%	93.73%	92.50%
Decision Tree	98.66%	98.91%	98.81%	94.52%	93.32%	93.92%	92.65%
Multi-Layer Perceptron	98.84%	98.96%	98.90%	94.83%	94.20%	94.51%	93.41%

TABLE 8. The performances of the classifiers given in Section IV-D in terms of recall, specificity, precision, f-measure, and mcc values. This time, contrary to Table 7, one month was used to train the respective model and the following month was used to test the performance of the model. The last row for each classifier gives the weighted average of the performance metrics for the tests conducted.

		Non-Fraud(%)			Fraud(%)			
Algorithm	**Month**	**Precision**	**Specificity**	**F-Measure**	**Precision**	**Recall**	**F-Measure**	**MCC**
Naive Bayes	Feb.	99.47	92.73	95.98	72.96	97.56	83.49	80.87
	Mar.	99.41	92.85	96.01	73.22	97.25	83.54	80.89
	Apr.	99.37	93.80	96.51	75.87	97.03	85.16	82.67
	May.	99.09	93.34	96.13	74.33	95.75	83.69	80.88
	Jun.	98.73	93.64	96.12	74.89	94.05	83.39	80.35
	Jul.	98.83	92.72	95.68	72.33	94.54	81.96	78.80
	Aug.	99.06	92.60	95.72	72.29	95.65	82.35	79.36
	W.Avg.	**99.21**	**93.16**	**96.08**	**73.92**	**96.30**	**83.63**	**80.86**
Random Forest	Feb.	98.65	98.88	98.76	94.37	93.26	93.81	92.58
	Mar.	98.39	99.17	98.78	95.69	91.93	93.77	92.57
	Apr.	98.64	99.19	98.92	95.87	93.19	94.51	93.44
	May.	98.31	99.15	98.72	95.57	91.51	93.50	92.25
	Jun.	98.07	99.02	98.54	94.89	90.33	92.55	91.13
	Jul.	97.94	98.78	98.36	93.68	89.70	91.65	90.04
	Aug.	97.73	98.77	98.25	93.58	88.65	91.05	89.35
	W.Avg.	**98.35**	**99.04**	**98.70**	**95.07**	**91.76**	**93.38**	**92.10**
Decision Tree	Feb.	98.59	98.85	98.72	94.22	92.96	93.59	92.31
	Mar.	98.65	98.92	98.78	94.54	93.28	93.91	92.69
	Apr.	98.56	99.09	98.83	95.34	92.81	94.06	92.90
	May.	98.09	99.21	98.65	95.85	90.39	93.04	91.75
	Jun.	98.14	98.95	98.54	94.55	90.69	92.58	91.15
	Jul.	97.71	98.55	98.12	92.45	88.50	90.43	88.59
	Aug.	97.45	98.66	98.05	92.91	87.18	89.95	88.07
	W.Avg.	**98.32**	**98.94**	**98.63**	**94.56**	**91.58**	**93.04**	**91.69**
Multi-Layer Perceptron	Feb.	98.75	98.82	98.78	94.09	93.77	93.93	92.71
	Mar.	98.49	99.19	98.84	95.84	92.43	94.10	92.97
	Apr.	98.70	99.14	98.92	95.60	93.52	94.55	93.48
	May.	98.12	99.29	98.70	96.25	90.54	93.31	92.07
	Jun.	98.25	98.93	98.59	94.53	91.25	92.86	91.47
	Jul.	97.98	98.73	98.35	93.44	89.88	91.63	90.00
	Aug.	96.99	98.76	97.87	93.23	84.81	88.82	86.83
	W.Avg.	**98.36**	**99.03**	**98.69**	**95.02**	**91.78**	**93.36**	**92.09**

instances were split up into train and validation subsets with a ratio of 7:3. The last row in the table for each classifier gives the weighted average of the performance metrics for the tests conducted.

A. SPATIAL ANALYSIS

In this section, we examined the effect of designing classification models according to card types, amount spent and characteristics of non-fraudulent and fraudulent transactions.

1) CARD TYPE BASED CLUSTERING

In this subsection, we explored the effect of spending characteristics for each card type (debit, classic, gold or business) on the fraud detection success. First of all, we obtained the distribution of the recall values of the selected classifiers in terms of card types, as depicted in Figure 5. Thus, before generating an independent classification model for each card type, we tried to establish a base line for performance comparison and obtained four new recall values for each classifier representing each card type. It is evident from Figure 5 that all the classifiers perform poorly for debit cards.

As the next step, we decided to design two card-type based scenarios, namely Scenario 1 and Scenario 2, using the selected classifiers. In Scenario 1, all the fraudulent transactions were used alongside with non-fraudulent transactions belonging only to the respective card type. On the other hand, in Scenario 2, both fraudulent and non-fraudulent instances were chosen based on the card type. For both scenarios, the aforementioned non-fraud to fraud ratio of 5:1 was

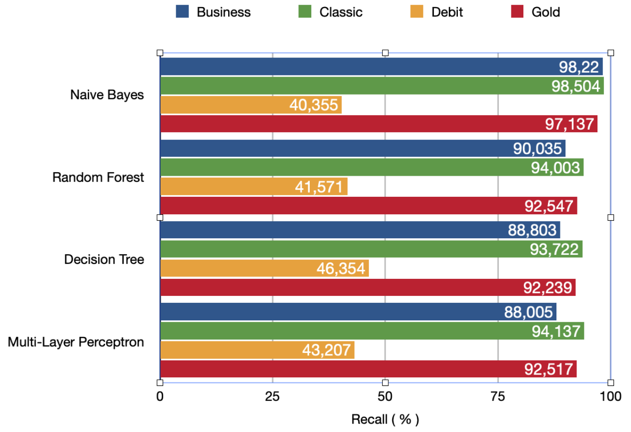


FIGURE 5. The distribution of correctly classified fraud instances according to card types.

TABLE 9. The overall number of non-fraud and fraud instances used in Scenario 1 and Scenario 2.

	Scenario 1		Scenario 2	
Card Type	**Non-fraud**	**Fraud**	**Non-fraud**	**Fraud**
Business	1170k	234k	55k	11k
Classic	1170k		720k	144k
Debit	1170k		35k	7k
Gold	1170k		350k	70k

TABLE 10. The overall number of non-fraud and fraud instances used in Scenario 1 and Scenario 2.

	Scenario 1		Scenario 2	
Amount Range	**Non-fraud**	**Fraud**	**Non-fraud**	**Fraud**
0-1	1170k	234k	780	156
1-10	1170k		10.5k	2.1k
10-100	1170k		640k	128k
100-1000	1170k		475k	95k
1000-10000	1170k		37k	7.4k
10000-100000	1170k		1.1k	227

preserved. Table 9 gives the overall number of non-fraud and fraud instances used in Scenario 1 and Scenario 2.

A comparative summary of the performances based on the recall value for all the selected classifiers are given in Figure 6. It is evident that except for the case of debit cards, card-based profiling does not help to boost the classification performance independent of the scenario used. For the debit card case both Naive Bayes and Random Forest classifiers yield the highest boost of about 26% points followed by Decision Tree and Multi-Layer Perceptron.

2) AMOUNT BASED CLUSTERING

In order to assess the effect of spending amounts on the fraud detection performance, we opted to group the transactions into bins using a logarithmic scale. The details of this logarithmic binning are given in Table 3.

A procedure similar to the procedure described in Section V-A1 was used to generate the models for each bin but this time amount ranges were used instead of card types. The overall number of non-fraud and fraud instances used in Scenario 1 and Scenario 2 are given in Table 10.

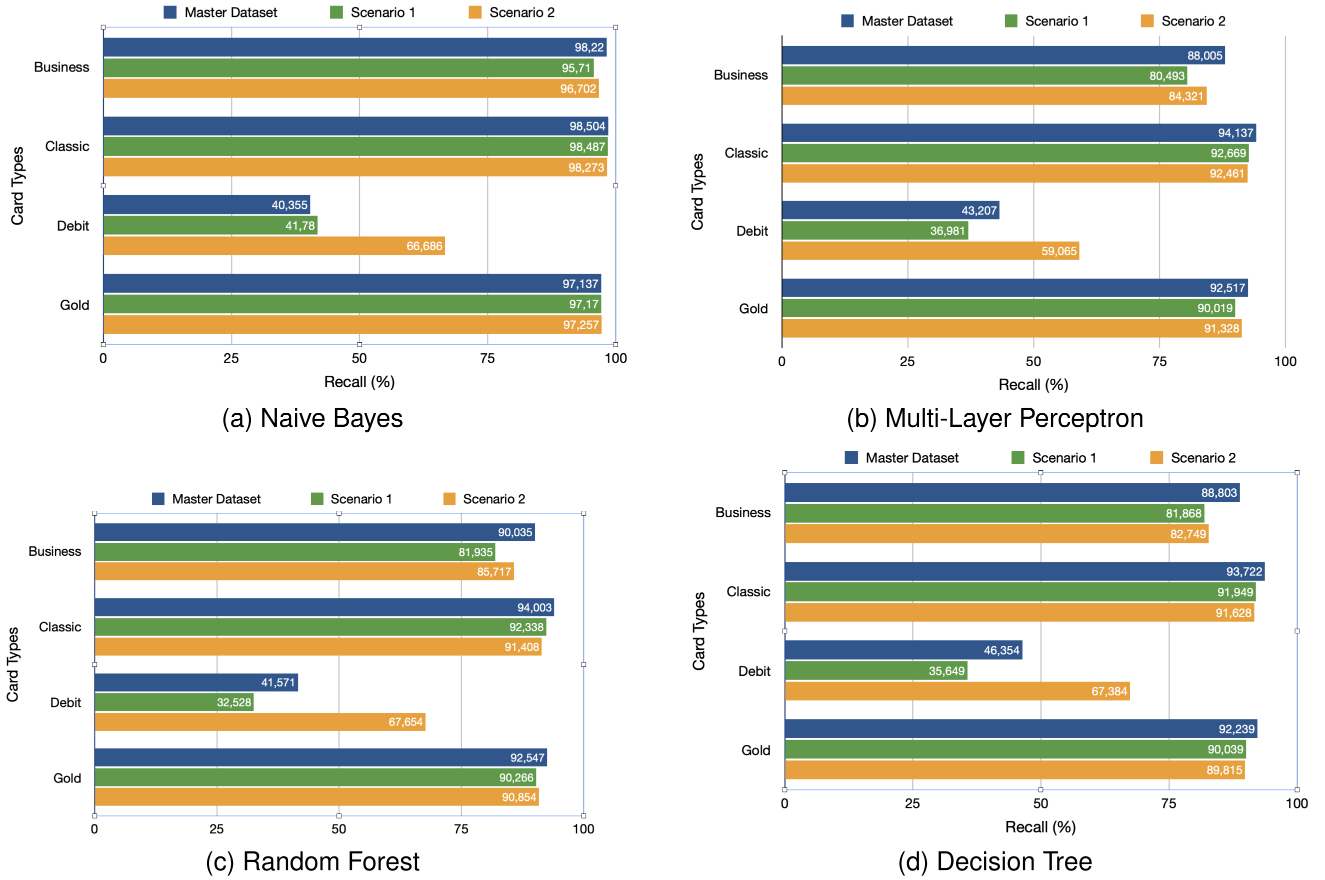
Figure 7 depicts the performance comparison of the generated models. Except for the Naive Bayes based classifiers, we believe that amount-based profiling has some merit which could be used to detect and prevent frauds with larger amounts as early as possible. Most of the time fraudsters attempt to exploit a stolen credit card information with amounts in the range of either 0 to 10 before committing a fraud with a considerable larger amount. Both Random Forest and Multi-Layer Perceptron classifiers show significant improvement over the master model for the aforementioned amount range. Therefore, an intelligent fraud detection system could decide to use the most appropriate detection model and boost performance. On the other hand, the same two classifiers also show better results for the 1000 to 10000 amount range. This amount range is the mostly exploited one after successfully committing a test fraud within the 0 to 10 amount range. Thus, this approach could be both used to detect and prevent larger frauds.

3) TRANSACTION BASED CLUSTERING

In order to observe the effect of grouping transactions according to their characteristics, non-fraudulent and fraudulent transactions were clustered using the k-Prototypes algorithm. The Elbow method was chosen to determine the number of clusters. The data set was divided into k clusters where k was chosen in the range of 1 to 200. For each iteration k value was increased by 5 and the Mean Square Error (MSE) [54] values were calculated. Figure 9 depicts the relationship between k and respectively calculated MSE values. According to the results, we opted to use 60 clusters for fraudulent transactions and 120 clusters for non-fraudulent transactions.

Since some clusters show similar characteristics, such clusters were merged according to the distances of their centroids iteratively. As a result of this merging operation, seven cases for non-fraudulent transactions with cluster sizes of (1, 20, 40, .., 120) and four different cases of fraudulent transactions with cluster sizes of (1, 20, 40, 60) were obtained. Afterwards, performance evaluation tests were run on the resulting 28 clustering combinations. For each scenario, the respective models' number of outputs corresponded to the number of total clusters. The output of the model was then mapped to a binary classification, consisting of fraud and non-fraud classes, by aggregating the results according to their cluster belonging. Then, the recall values for each scenario was calculated based upon aggregation which are given in Figure 8.

The results in the figure clearly show that Random Forest, Decision Tree, and Multi-Layer Perceptron classifiers attain almost the same performance regardless how the non-fraudulent and fraudulent clusters are crossed. On the other hand, Naive Bayes classifier is very susceptible to this setup as Figure 8-a demonstrates a crossing of an unequal number of non-fraudulent and fraudulent clusters produce significantly worse results. For Naive Bayes classifier, to obtain optimum results the number of clusters for



(a) Naive Bayes

(b) Multi-Layer Perceptron

(c) Random Forest

(d) Decision Tree

FIGURE 6. **A comparative summary of the performances based on the recall value for all the selected classifiers for card-type based scenarios.**

both classes should be equal. Also, having the non-fraudulent instances unclustered in the classification process suppresses this phenomenon and the Naive Bayes classifier produces almost the same result regardless of the number of fraudulent clusters.

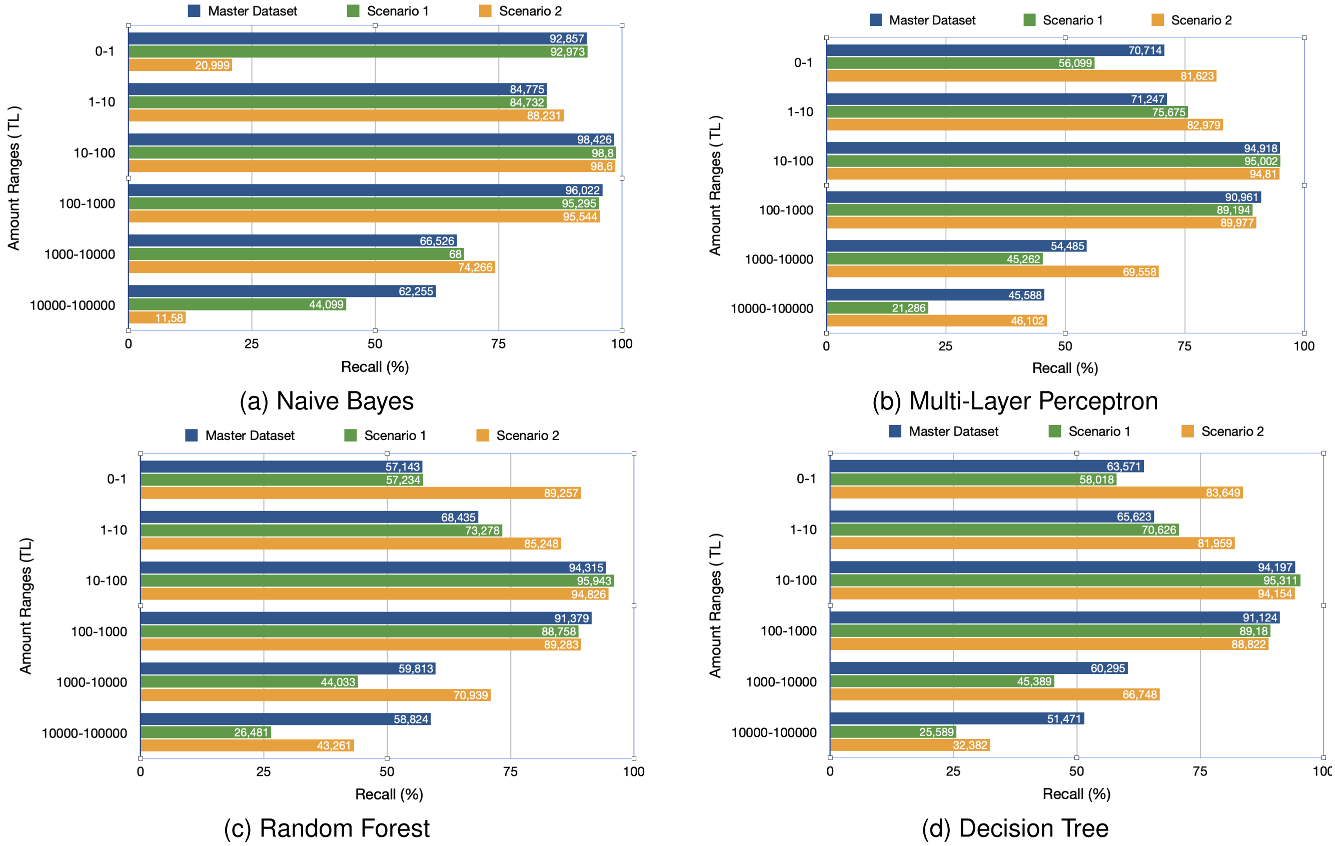
B. TEMPORAL ANALYSIS

In this subsection, we examined the temporal validity of a generated model, the effect of temporal changes within the instances on the performance of the classification as well as the response of the generated model in case of previously unencountered fraudulent activities, which practically corresponds to zero-day attack analysis. Although temporal analysis and zero-day attack analysis of a given model could be perceived to be the same, there is a clear distinction how these analyses were performed. In our temporal analysis tests, a given model trained with all the types of fraudulent instances of up to 6 consecutive months was tested using instances from the upcoming month to investigate the effects of the drift in the data. On the other hand, in the zero-day attack analysis fraud instances were clustered based on their transactional characteristics and some clusters were specifically left out from the training data set, which we refer to as never before-seen fraud attacks.

1) DETERIORATION RATE OF DETECTION MODEL

In this subsection, the deterioration rate of classification success was examined as the fraud detection model was used to detect fraudulent transactions farther away from the time of the training and validation dataset. By taking into account the considerably less number fraudulent instances in August 2017, we have chosen March 2017 as the middle point within our dataset so that we were able to test the temporal change of the success of a given model up to 2 months back and 5 months forward. Also, March 2017 has the most number of fraudulent instances so this was another point of consideration for making it the pivot month.

Figure 10 shows a decay in recall rates as we move away from March independent of the classifier used. The decay is significantly smaller for the Naive Bayes classifier in comparison with the other classifiers. On the other hand, specificity was not affected at all. For a financial institution, these findings dictate that any given fraud detection model should be updated with the instances of the current month to be prepared for the upcoming month. Also, ideally with enough processing power the model could be refreshed on a weekly even daily basis within the present month.



(a) Naive Bayes

(b) Multi-Layer Perceptron

(c) Random Forest

(d) Decision Tree

FIGURE 7. **The comparison of recall values of the amount-range based scenarios, Scenario 1 and Scenario 2 against the master model.**

2) EFFECT OF TIME SPAN OF TRAIN SET INSTANCES

In this subsection, we concentrated on the effect of temporal changes within the instances on the classification performance. Therefore, as previously stated because of the lack of sufficient number of fraudulent instances, we left out August 2017 and chose July 2017 as our test data set. Then, starting with June 2017 and going back to January 2017 we generated at total of 6 models for each selected classifier by including another previous month to the already included ones. For each model, the respective data set was split up in itself 70% for training 30% for validation. The obtained performance results in terms precision, recall, specificity, and f-measure for each selected classifier are summarized in Table 11.

Except for the case of the Multi-Layer Perceptron, the inclusion of past instances does not contribute to the performance of a given classifier. For Multi-Layer Perceptron, with some fluctuations, up to 2% points gain was observed.

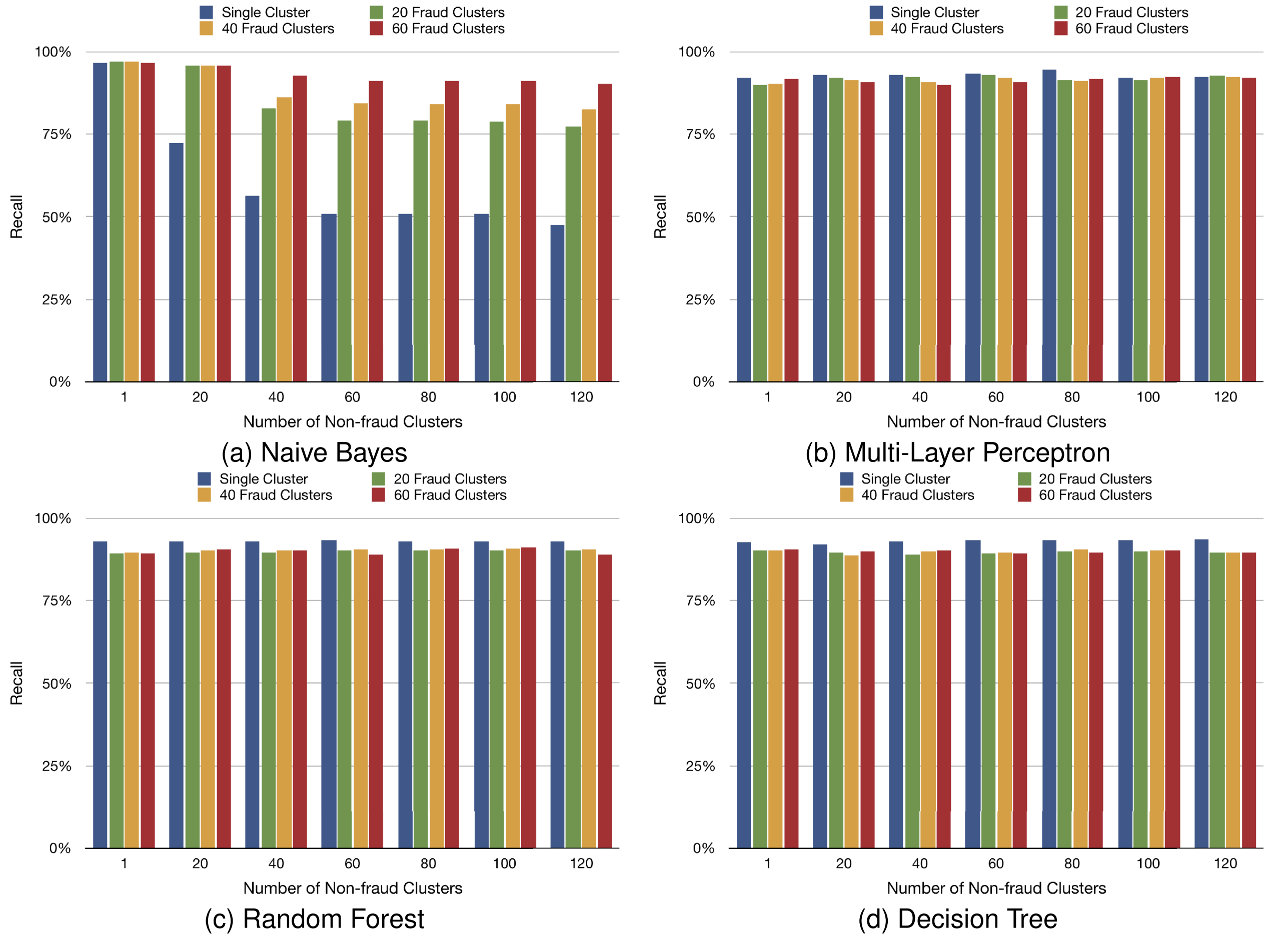
3) ZERO-DAY PERFORMANCE

Although temporal analysis and zero-day attack analysis of a given model could be perceived to be the same, there is a clear distinction how these analyses were performed. In our temporal analysis tests, a given model trained with all the types of fraudulent instances of up to 6 consecutive months was tested using instances from the upcoming month to investigate the effects of the drift in the data. On the other hand, in the zero-day attack analysis fraud instances were clustered based on their transactional characteristics and some clusters were specifically left out from the training data set, which we refer to as never before-seen fraud attacks.

In financial fraudulent transaction detection, we define the zero-day performance of a detection model as its response to a fraudulent activity with a type not included in the training data set during model generation. Therefore, we deliberately excluded some types of fraudulent transactions from the training data set and then tested the performance of the generated models with those previously unseen instances. We carried out zero-day performance tests on the master data set as well as data sets generated for the analysis of card type-based profiling and amount-based profiling. Figure 11, shows the results of the performed tests for each selected classifier in terms of recall values. Again, Naive Bayes classifier demonstrated a better performance than the other classifiers. Nevertheless, we deem the performance of all classifiers acceptable for this previously unexplored challenging task.

C. MONETARY PERFORMANCE ANALYSIS OF FRAUD DETECTION

To the best of our knowledge, fraud detection studies conducted so far have assessed their performance in terms of accuracy, recall, precision, and f-measure values derived



(a) Naive Bayes

(b) Multi-Layer Perceptron

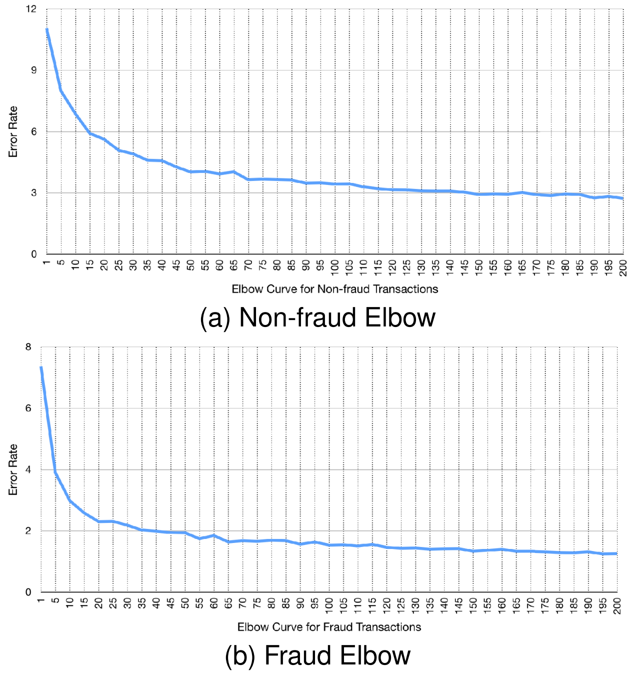
(c) Random Forest

(d) Decision Tree

FIGURE 8. The performances of the selected classifiers in terms of recall value for transaction-based scenario.

TABLE 11. The performance summary of the selected classifiers to reflect their behaviour against temporal changes in the instances used for training and validation.

Algorithm	Train Period	Non-Fraud Precision	Non-Fraud Recall	Non-Fraud F-Measure	Fraud Precision	Fraud Recall	Fraud F-Measure
Naive Bayes	Jan.-Jun.	98.83	92.99	95.82	73.07	94.54	82.43
	Feb.-Jun.	98.82	93.03	95.84	73.17	94.47	82.47
	Mar.-Jun.	98.81	93.04	95.84	73.19	94.42	82.46
	Apr.-Jun.	98.81	92.97	95.80	73.01	94.43	82.35
	May.-Jun.	98.82	92.93	95.78	72.89	94.47	82.29
	Jun.-Jun.	98.83	92.72	95.68	72.33	94.54	81.96
Random Forest	Jan.-Jun.	97.71	98.91	98.31	94.26	88.49	91.28
	Feb.-Jun.	97.87	98.92	98.39	94.32	89.31	91.75
	Mar.-Jun.	97.85	99.00	98.42	94.71	89.18	91.86
	Apr.-Jun.	97.95	98.92	98.43	94.37	89.72	91.99
	May.-Jun.	97.46	98.85	98.15	93.84	87.20	90.40
	Jun.-Jun.	97.92	98.82	98.37	93.88	89.58	91.68
Decision Tree	Jan.-Jun.	97.81	98.56	98.18	92.54	89.03	90.75
	Feb.-Jun.	97.62	98.78	98.20	93.58	88.05	90.74
	Mar.-Jun.	97.71	98.71	98.20	93.23	88.49	90.80
	Apr.-Jun.	97.79	98.66	98.23	93.06	88.94	90.95
	May.-Jun.	97.64	98.75	98.19	93.41	88.14	90.70
	Jun.-Jun.	97.70	98.54	98.12	92.44	88.49	90.42
Multi-Layer Perceptron	Jan.-Jun.	97.98	99.10	98.54	95.25	89.87	92.48
	Feb.-Jun.	97.68	99.18	98.42	95.59	88.27	91.79
	Mar.-Jun.	97.80	99.15	98.47	95.45	88.90	92.06
	Apr.-Jun.	98.18	98.98	98.58	94.74	90.88	92.77
	May.-Jun.	97.40	99.22	98.31	95.75	86.86	91.09
	Jun.-Jun.	97.24	99.14	98.18	95.27	86.04	90.42



(a) Non-fraud Elbow

(b) Fraud Elbow

FIGURE 9. The relationship between the number of clusters and corresponding MSE values for both non-fraudulent and fraudulent transactions.

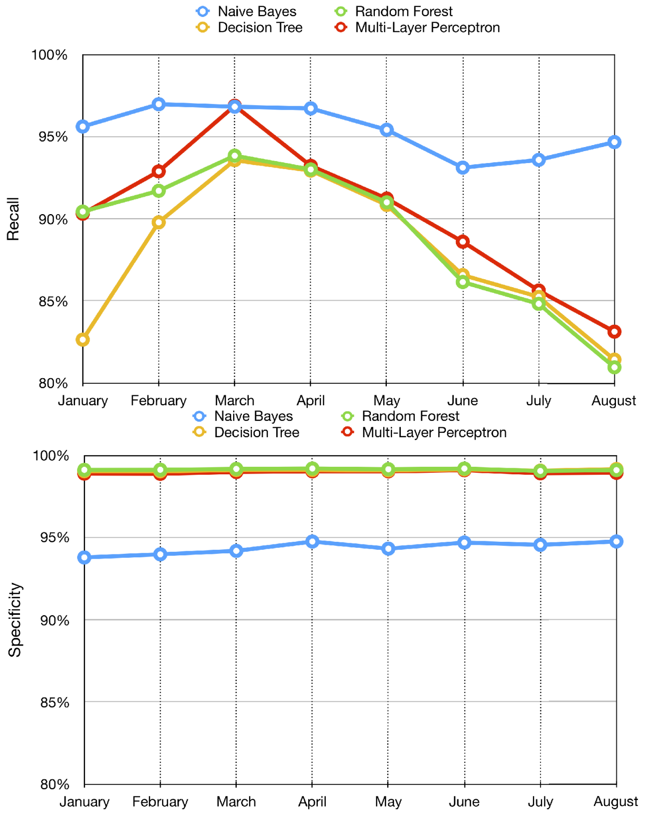


FIGURE 10. The time durability of the master model in terms of recall and specificity over 8 months.

from correctly and incorrectly classified transactions without considering the financial value of the respective transactions [33]. In this subsection, we evaluated the performance of the selected classifiers in terms of the monetary value they represent.

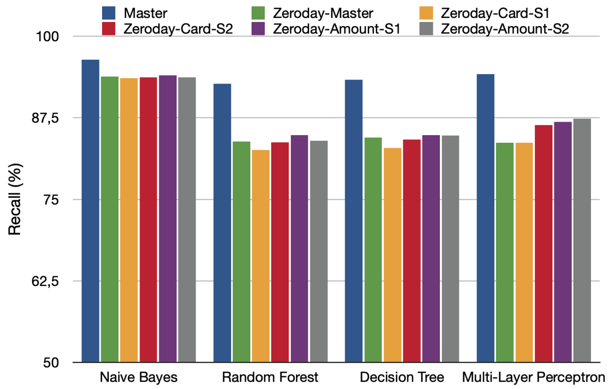


FIGURE 11. The resiliency of the master model in terms of recall for zero day attacks.

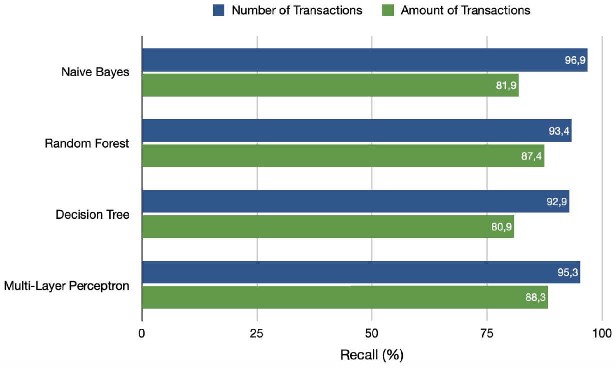


FIGURE 12. The green bars represent the monetary equivalent of the correctly classified fraud transactions divided to the monetary value corresponding to the total number of fraud transactions for each classifier.

For this analysis, we opted to use the master data set and we based our evaluation on the recall values. For each classifier, the monetary equivalent of the correctly classified fraud transactions were divided to the monetary value corresponding to the total number of fraud transactions. The so obtained ratios and the recall values of the respective classifier are given in Figure 12. Our intention was to transform the recall value obtained from instances having the same weight into a monetary recall value representing what percentage of financial loss would be recovered with the use of that classifier.

A closer look into the Figure 12 reveals that even though Random Forest and Multi-Layer Perceptron classifiers were outperformed by the Naive Bayes classifier, both of them were able to represent a higher financial percentage of the overall fraudulent transactions. It is our opinion to pursue this way of investigation in a future study to build more formal relationship between the plain performance metrics and the so-called financial performance metrics.

D. TIME ANALYSIS

In banking transactions, it is crucial that the payments are not interrupted [55]. Therefore the running time performance of

TABLE 12. **The time required for each classifier to classify a given transaction.**

Algorithm	Time (ms)
Naive Bayes	0.52
Random Forest	5.54
Decision Tree	0.06
Multi-Layer Perceptron	1.43

the system was also analyzed. IBM POWER9 based servers were used for the timing tests. The servers had a total of 160 CPU cores and a total of 1.1 TB of memory running Ubuntu 16.04 operating system. BKM processes almost 15 million transactions on a daily basis which corresponds roughly to 174 transactions per second. For a transaction to go through the system without further delay the processing time of any given transaction should be less than 6 milliseconds. Under these constraints, to evaluate the timing performance of our classification models, we ran the respective classifiers with 1 million instances per classifier. The average pre-processing time was determined to be 0.8 milliseconds independent of the classifier, whereas the average run times for each classifier is given in Table 12. The results show that our generated models with the exception of Random Forest classifier could be deployed on a single server and still be within safe limits in terms of pre-processing and classification time. For the Random Forest based model, a simple load balancing scheme should suffice to meet the aforementioned timing constraints.

VI. CONCLUSION

In this study, we completed a comprehensive analysis on the biggest data set, namely BKM data set, ever used in fraud detection domain. It contains 4 billions non-fraud and 245k fraud transactions contributed to by the 35 banks in Turkey. Unlike most of the research work cited in the literature, we chose to generate fraud detection models according to predefined profile types. Those profile types were based on card-type, amount-range and the characteristics of the financial transactions. We showed that except for the case of debit cards, card-based profiling does not help to boost the classification performance independent of the scenario used. As for the amount-based profiling both Random Forest and Multi-Layer Perceptron classifiers showed a significant improvement over the master model for the 0 to 10 and 1000 to 100000 amount range in Turkish Lira. Therefore, an intelligent fraud detection system could decide to use the most appropriate detection model and boost performance. As well as generating multiple detection models based on the transaction characteristics reversely affected the model's performance. This is largely due to the fact that the imbalance between non-fraudulent and fraudulent instances gets more dominant with the lesser number of fraudulent instances. On the other hand, the resilience of the detection models is strongly related to the number of instances and their time span. Therefore, our test results show that any fraud detection model should be periodically updated in order to include recent fraudulent instances. Regarding the zero-day performance, we showed that all models without exception demonstrated some weakness against previously unencountered fraudulent activities. Nonetheless, the overall performance of the classifiers were acceptable for such cases. As another contribution, we expressed the performance of a classifier in terms of the financial value it represents. For the BKM data set our experiments showed that large number of false positives do not necessarily correspond to a large financial loss. We think that this observation begs for more detailed investigation in future studies.

ACKNOWLEDGEMENT

The authors would like to sincerely thank Bankalararası Kart Merkezi (BKM) who prepared the data set and helped the authors to understand the fraudulent behaviours and data set features as well as provided the necessary infrastructure for GPU-based machine learning on big data. They also would like to thank Taner GUVEN, Ph.D. student at the Department of Computer Engineering, Yildiz Technical University, for his great efforts to accomplish additional tests for this study.

REFERENCES

[1] RBR. (2018). *Global Payment Cards Data and Forecasts to 2023*. [Online]. Available: https://www.rbrlondon.com/research/global-cards/

[2] losspreventionmedia. (2018). *The Latest Credit Card Fraud Statistics and Insights*. [Online]. Available: https://losspreventionmedia.com/credit-card-fraud-statistics-and-insights/

[3] TBB. (2019). *Detection and Prevention Methods of Fraud in Banking*. [Online]. Available: https://www.tbb.org.tr/gec/KTPV14.pdf

[4] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90–113, Jun. 2016.

[5] G. Potamitis, "Design and implementation of a fraud detection expert system using ontology-based techniques," Univ. Manchester, Manchester, U.K., Tech. Rep. COMP60990, 2013.

[6] N. Laleh and M. A. Azgomi, "A taxonomy of frauds and fraud detection techniques," in *Information Systems, Technology and Management. ICISTM* (Communications in Computer and Information Science), vol. 31, S. K. Prasad, S. Routray, R. Khurana, and S. Sahni, Eds. Berlin, Germany: Springer, 2009, doi: 10.1007/978-3-642-00405-6_28.

[7] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in *Proc. 11th Int. Conf. Tools Artif. Intell.*, 1999, pp. 103–106.

[8] P. Gosset and M. Hyland, "Classification, detection and prosecution of fraud in mobile networks," in *Proc. ACTS Mobile Summit*, Sorrento, Italy, 1999, pp. 1–6.

[9] L. Cortesão, F. Martins, A. Rosa, and P. Carvalho, "Fraud management systems in telecommunications: A practical approach," in *Proc. ICT*, 2005, pp. 1–5.

[10] M. K. Sparrow, *License to Steal: How Fraud Bleeds America's Health Care System*. Abingdon, U.K: Routledge, 2019.

[11] R. M. Musal, "Two models to investigate medicare fraud within unsupervised databases," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8628–8633, Dec. 2010.

[12] S. Chen and A. Gangopadhyay, "A novel approach to uncover health care frauds through spectral analysis," in *Proc. IEEE Int. Conf. Healthcare Informat.*, Sep. 2013, pp. 499–504.

[13] L. Šubelj, Š. Furlan, and M. Bajec, "An expert system for detecting automobile insurance fraud using social network analysis," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 1039–1052, Jan. 2011.

[14] C.-H. Wen, M.-J. Wang, and L. W. LAN, "Discrete choice modeling for bundled automobile insurance policies," *J. Eastern Asia Soc. Transp. Stud.*, vol. 6, pp. 1914–1928, 2005.

[15] W.-H. Chang and J.-S. Chang, "An effective early fraud detection method for online auctions," *Electron. Commerce Res. Appl.*, vol. 11, no. 4, pp. 346–360, Jul. 2012.

[16] D. H. Chau, S. Pandit, and C. Faloutsos, ''Detecting fraudulent personalities in networks of online auctioneers,'' in *Knowledge Discovery in Databases: PKDD* (Lecture Notes in Computer Science), vol. 4213, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin, Germany: Springer, 2006, doi: 10.1007/11871637_14.

[17] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, ''Credit card fraud detection using Bayesian and neural networks,'' in *Proc. 1st Int. Naiso Congr. Neuro Fuzzy Technol.*, 2002, pp. 261–270.

[18] T. Cody, S. Adams, and P. A. Beling, ''A utilitarian approach to adversarial learning in credit card fraud detection,'' in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 237–242.

[19] C. Hines and A. Youssef, ''Machine learning applied to rotating check fraud detection,'' in *Proc. 1st Int. Conf. Data Intell. Secur. (ICDIS)*, Apr. 2018, pp. 32–35.

[20] A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, ''Feature engineering strategies for credit card fraud detection,'' *Expert Syst. Appl.*, vol. 51, pp. 134–142, Jun. 2016.

[21] O. S. Yee, S. Sagadevan, and N. H. A. H. Malim, ''Credit card fraud detection using machine learning as data mining technique,'' *J. Telecommun., Electron. Comput. Eng.*, vol. 10, nos. 1–4, pp. 23–27, 2018.

[22] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, ''Credit card fraud detection: A realistic modeling and a novel learning strategy,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.

[23] C. Hines and A. Youssef, ''Machine learning applied to point-of-sale fraud detection,'' in *Machine Learning and Data Mining in Pattern Recognition. MLDM* (Lecture Notes in Computer Science), vol. 10934, P. Perner, Ed. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-96136-1_23.

[24] P. H. Tran, K. P. Tran, T. T. Huong, C. Heuchenne, P. HienTran, and T. M. H. Le, ''Real time data-driven approaches for credit card fraud detection,'' in *Proc. Int. Conf. E-Business Appl. (ICEBA)*, 2018, pp. 6–9.

[25] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, ''Deep learning detecting fraud in credit card transactions,'' in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 129–134.

[26] R. Saia, ''A discrete wavelet transform approach to fraud detection,'' in *Network and System Security. NSS* (Lecture Notes in Computer Science), vol. 10394, Z. Yan, R. Molva, W. Mazurczyk, and R. Kantola, Eds. Cham, Switzerland: Springer, 2017, 10.1007/978-3-319-64701-2_34.

[27] R. Saia and S. Carta, ''A Frequency-domain-based pattern mining for credit card fraud detection,'' in *Proc. 2nd Int. Conf. Internet Things, Big Data Secur.*, 2017, pp. 386–391.

[28] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, ''GMM-based undersampling and its application for credit card fraud detection,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[29] H. Wu and G. Liu, ''A hybrid model on learning cross features for transaction fraud detection,'' in *Proc. ICDM*, 2019, pp. 88–102.

[30] L. Zheng, G. Liu, C. Yan, and C. Jiang, ''Transaction fraud detection based on total order relation and behavior diversity,'' *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 3, pp. 796–806, Sep. 2018.

[31] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, ''Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism,'' *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3637–3647, Oct. 2018.

[32] E. Kim, J. Lee, H. Shin, H. Yang, S. Cho, S.-K. Nam, Y. Song, J.-A. Yoon, and J.-I. Kim, ''Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning,'' *Expert Syst. Appl.*, vol. 128, pp. 214–224, Aug. 2019.

[33] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, ''Random forest for credit card fraud detection,'' in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Mar. 2018, pp. 1–6.

[34] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, ''Credit card fraud detection using AdaBoost and majority voting,'' *IEEE Access*, vol. 6, pp. 14277–14284, 2018.

[35] 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,'' *Inf. Sci.*, 2019, doi: 10.1016/j.ins.2019.05.042.

[36] R. J. Bolton *et al.*, ''Unsupervised profiling methods for fraud detection,'' in *Credit Scoring and Credit Control VII*. Citeseer, 2001, pp. 235–255.

[37] A. Pumsirirat and L. Yan, ''Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 1, pp. 18–25, 2018.

[38] R. Saia and S. Carta, ''Evaluating credit card transactions in the frequency domain for a proactive fraud detection approach,'' in *Proc. 14th Int. Joint Conf. e-Business Telecommun.*, 2017, pp. 335–342.

[39] C. Elkan, ''The foundations of cost-sensitive learning,'' in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 17, no. 1. New Jersey, NJ, USA: Lawrence Erlbaum Associates, 2001, pp. 973–978.

[40] D. J. Hand, C. Whitrow, N. M. Adams, P. Juszczak, and D. Weston, ''Performance criteria for plastic card fraud detection tools,'' *J. Oper. Res. Soc.*, vol. 59, no. 7, pp. 956–962, Jul. 2008.

[41] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, ''Cost sensitive credit card fraud detection using bayes minimum risk,'' in *Proc. 12th Int. Conf. Mach. Learn. Appl.*, Dec. 2013, pp. 333–338.

[42] Y. K. Jain and S. K. Bhandare, ''Min max normalization based data perturbation method for privacy protection,'' *Int. J. Comput. Commun. Technol.*, vol. 2, no. 8, pp. 45–50, Oct. 2011.

[43] Y. Yang and J. O. Pedersen, ''A comparative study on feature selection in text categorization,'' in *Proc. ICML*, 1997, vol. 97, nos. 412–420, p. 35.

[44] A. G. Karegowda, A. S. Manjunath, and M. A. Jayaram, ''Comparative study of attribute selection using gain ratio and correlation based feature selection,'' *Int. J. Inf. Technol. Knowl. Manage.*, vol. 2, no. 2, pp. 271–277, 2010.

[45] D. Morariu, R. Cretulescu, and M. Breazu, ''Feature selection in document classification,'' in *Proc. 4th Int. Conf. Romania Inf. Sci. Inf. Literacy (ISSN-L)*, 2013, pp. 0255–2247.

[46] K. Kira and L. A. Rendell, ''The feature selection problem: Traditional methods and a new algorithm,'' in *Proc. AAAI*, vol. 2, 1992, pp. 129–134.

[47] L. Yu and H. Liu, ''Feature selection for high-dimensional data: A fast correlation-based filter solution,'' in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 856–863.

[48] Z. Huang, ''Extensions to the K-means algorithm for clustering large data sets with categorical values,'' *Data Mining Knowl. Discovery*, vol. 2, no. 3, pp. 283–304, Sep. 1998.

[49] P. Bholowalia and A. Kumar, ''EBK-means: A clustering technique based on elbow method and k-means in WSN,'' *Int. J. Comput. Appl.*, vol. 105, no. 9, 2014.

[50] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[51] L. Breiman, ''Random forests,'' *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[52] A. McCallum and K. Nigam, ''A comparison of event models for naive bayes text classification,'' in *Proc. AAAI Workshop Learn. Text Categorization*, 1998, vol. 752, no. 1, pp. 41–48.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[54] Y. Ephraim and D. Malah, ''Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,'' *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.

[55] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, ''SCARFF: A scalable framework for streaming credit card fraud detection with spark,'' *Inf. Fusion*, vol. 41, pp. 182–194, May 2018.

BARIS CAN was born in İstanbul, Turkey, in 1995. He graduated from the Computer Engineering Department, Yildiz Technical University, Istanbul, in January 2019. He was a Junior Backend Developer with the Transport Mode Detection Project, Intelligent Systems Laboratory, Yildiz Technical University, in 2016. From July to September 2017, he was an Intern with the TÜBİTAK BİLGEM, Turkey. From July to September 2017, he was a Junior Mobile Developer with the Cloud Processing Unit. Since February 2019, he has been working with the Interbank Card Center, Turkey. From October to January 2019, he was a Long-Term Intern with the Interbank Card Center. He currently works with the TÜBİTAK BİLGEM. He is also a Data Analyst with the Card Fraud Detection. His research interests include mobile computing, mobile technologies and applications, activity recognition, fraud detection, deep learning, cloud systems, and big data.

ALI GOKHAN YAVUZ received the Ph.D. degree in computer engineering from Yildiz Technical University, Istanbul, Turkey. He is currently an Associate Professor with the Department of Computer Engineering, Yildiz Technical University. He is also the Co-Director of the Intelligent Systems Laboratory. His current research interests include systems and network security, cloud computing, and big data.

ELIF M. KARSLIGIL received the Ph.D. degree in computer engineering from Yildiz Technical University, Istanbul, Turkey. She was a Postdoctoral Researcher with the Intelligent Communication Laboratory, NTT-Japan, in 2002. She is currently an Associate Professor with the Department of Computer Engineering, Yildiz Technical University. She is also the Co-Director of the Intelligent Systems Laboratory. Her current research interests include computer vision, machine learning, and deep learning.

M. AMAC GUVENSAN (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer engineering from Yildiz Technical University, Istanbul, Turkey, in 2002, 2006, and 2011 respectively. From 2009 to 2010, he visited the Wireless Networks and Embedded Systems Laboratory, University at Buffalo, The State University of New York. He is currently an Assistant Professor with the Department of Computer Engineering, Yildiz Technical University. He is a member of Intelligent Systems Laboratory, Yildiz Technical University. His current research interests include pervasive and ubiquitous computing, mobile technologies and applications, intelligent transportation systems, machine learning, and the Internet of Things.