ATLANTIS PRESS

Atlantis Studies in Uncertainty Modelling, volume 1

11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)

Credit card fraud detection by dynamic incremental semi-supervised fuzzy clustering

Gabriella Casalino, Giovanna Castellano and Corrado Mencar
Department of Computer Science, University of Bari Aldo Moro, Italy,
{gabriella.casalino, giovanna.castellano, corrado.mencar}@uniba.it

Abstract

The problem of credit card fraud detection is approached by a semi-supervised classification task on a data stream. The DISSFCM algorithm is applied, which is based on Dynamic Incremental Semi-Supervised Fuzzy C-Means that processes data grouped in small-size chunks. Experimental results on a real-world dataset of credit card transactions show that DISSFCM has comparable results with some fully-supervised stream data classification methods, also in presence of a high percentage of unlabeled data.

Keywords: Credit card fraud detection; Data stream classification; Semi-supervised fuzzy clustering; Incremental adaptive learning.

1 Introduction

Financial frauds are exponentially growing in the last decades due to money dematerialization. Credit cards are preferred to cash for any kind of payments, either online or offline [1]. Consequently, credit card fraud is becoming an emerging problem, since it annually causes the loss of billions of dollars. *Fraud Prevention Systems* (FPS), such as encryption algorithms, firewalls, etc., are used to stop frauds before they occur in the system. However these mechanisms are not sufficient to completely avoid frauds [2] and a further protection level is needed. *Fraud Detection Systems* (FDS) are used to discover and identify fraudulent activities that are already entered in the system.

The huge amount of information related to the credit card activities, makes manual analysis time consuming and inefficient, or even impractical. For this reason, automatic techniques are necessary to analyze the customer activity data flow. In this scenario, data mining techniques are widely used for credit fraud detection, as they are able to detect hidden patterns in data, related to the fraudulent activities. Indeed the data analysis process is based on the study of customers' behaviour in order to separate legitimate from fraudulent transactions. These methods construct models, based on the previous customer behavior, to detect anomalies in his/her activities.

Due to the dynamic nature of credit card transaction data, incremental learning, rather than static learning, is suitable for fraud detection. Incremental learning considers data as a continuous stream and processes each new instance on arrival [3]. In this context it is important to preserve the previously acquired knowledge as well as to properly update it as new observations arrive. When the data distribution changes, the classifier should be able to learn from a new fraud distribution and forget outdated knowledge. A fraud classifier is required to be able to respond to changes in the data distribution (concept drift), while ensuring that it still retains relevant past knowledge. As fraud evolves over time, the classification model should be adapted to the new distribution.

The recent literature of fraud detection includes both supervised and unsupervised techniques. Supervised methods create models to discriminate between known fraudulent and non-fraudulent transactions by requiring that class labels (e.g. genuine or fraudulent) of past transactions are available, so that new observations can be assigned to classes [4]. However, accurate labeling of fraudulent transactions in historical databases is an information that is often in short supply or non-existent.

Indeed, the assumption of labeled data is quite unrealistic in the context of credit card transactions [5]. In a real-world FDS, the stream of payment requests is quickly scanned by automatic tools that analyze all the transactions and alert the most suspicious. Automatic analysis is often not sufficiently reliable in such a delicate context, therefore intelligent data analysis [6],

Copyright © 2019, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

which involves a human domain expert in the analysis loop, could be required to decide when an activity is actually fraudulent . Thus, alerts are then inspected by professional investigators that contact the cardholders to determine the true nature (either genuine or fraudulent) of each alerted transaction. By doing this, investigators provide a feedback to the system in the form of labeled transactions, which can be used to train or update the classifier, in order to preserve (or possibly improve) the fraud-detection performance over time. Due to time and cost constraints, the vast majority of transactions cannot be verified by investigators and remain unlabeled until customers discover and report frauds, or until a sufficient amount of time has elapsed such that those transactions are considered genuine. Thus, in practice, most of labeled samples are provided with a substantial delay, a problem known as verification latency [7]. The only recent supervised information made available to update the classifier is provided through the alert-feedback interaction. Most papers in the literature using supervised learning ignore the verification latency as well as the alertfeedback interaction, and unrealistically assume that the label of each transaction is regularly made available to the FDS (e.g., on a daily basis) [8].

Supervised methods require that we have examples of both classes, and they can only be used to detect frauds of a type that has previously occurred. These methods also suffer from the problem of unbalanced class sizes: in fraud detection problems, the legitimate transactions generally far outnumber the fraudulent ones and this imbalance can cause misspecification of models.

Unsupervised methods do not require the prior knowledge on class transactions and are oriented to detect new fraudulent behaviours or unusual transactions [9, 10]. For example in [11] clustering is used to form customer profiles in order to identify new hidden fraud patterns. They model a baseline distribution that represents normal behaviour and then attempt to detect observations that show greatest departure from this norm. Hence unsupervised methods detect behavior anomalies by identifying transactions that do not conform to the majority. An advantage of using unsupervised methods over supervised methods is that previously undiscovered types of fraud may be detected with no need of labeled data on past transactions. Unsupervised learning is suitable when the behavior of fraudsters is evolving and we need to identify new patterns or anomalies (outliers). Hence previously undiscovered types of fraud may be detected. However, completely unsupervised methods are unable to exploit expert knowledge that can be injected by labeling transactions as fraud/non-fraud; this is crucial as anomalies may just be the initial manifestation of an evolution of a customer's behavior.

For this reason we believe that semi-supervised learning, using both unlabeled and labeled data, may be a good solution to design classifiers for fraud detection. Despite the large number of works on semi-supervised learning [12, 13], to our knowledge there are no papers that address the problem of credit card fraud detection by partially supervised learning.

In this paper we propose the use of incremental semi-supervised learning for credit card fraud detection. We show that our dynamic incremental semi-supervised fuzzy clustering algorithm (DISSFCM) [14], [15] can be an effective technique to recognize fraudulent activities in credit card transaction data streams.

2 Data stream classification by DISSFCM

The structure of the data stream X is formalized as a sequence of *chunks*, i.e. $X = X_1, X_2, \ldots, X_t, \ldots$. Each chunk $X_t \subset \mathbb{R}^n$ includes a number N_t of samples. Each sample $\mathbf{x} \in \mathbb{R}^n$ is characterized by numerical features and belongs to a class in $\mathcal{C} = \{1, \ldots, C\}$ through a classification function $f : X \mapsto \mathcal{C}$. According to the semi-supervised hypothesis, the classification function is generally known only for some samples. We formalize this hypothesis through a function $b : X \mapsto \{0, 1\}$ such that $b(\mathbf{x}) = 1$ iff $\mathbf{x}$ is *pre-labeled*, i.e. its class value $f(\mathbf{x})$ is known. This is a reasonable hypothesis in the context of credit card fraud detection (see Section 1).

The goal is to create a data stream classifier by exploiting partial supervision when available. To this aim, we leverage a fuzzy clustering process that takes into account the partial availability of class labels among samples. In [16] we introduced an incremental semi-supervised clustering method for data stream classification. The method was successively refined by enabling the dynamic determination of the number of clusters through a splitting procedure, leading to the DISSFCM (Dynamic Incremental Semi-Supervised FCM) algorithm [14].

One key feature of DISSFCM is the possibility to exploit partial supervision when available. Namely, when some pre-labeled data are available in a chunk, their labels can be used to drive the clustering process. The presence of pre-labeled data is not mandatory but it should be assured in the first chunk in order to properly initialize the cluster prototypes. The core of DISSFCM is the Semi-Supervised Fuzzy C-Means (SSFCM) algorithm [17], which embeds partial supervision in the standard FCM algorithm by adapting the

objective function as follows:

$$J = \sum_{k=1}^{K}\sum_{j=1}^{N_t} u_{jk}^m d_{jk}^2 + \alpha \sum_{k=1}^{K}\sum_{j=1}^{N_t} (u_{jk} - b_j f_{jk})^m d_{jk}^2 \quad (1)$$

where $K \geq C$ is the number of clusters, $N_t = |X_t|$ is the cardinality of the t-th chunk in the data stream, $u_{jk} \in [0,1]$ is the membership degree of a sample $\mathbf{x}_j$ in the k-th cluster, $m > 1$ is a *fuzzification* factor that is used in FCM to regulate the smoothness of fuzzy clusters (we use $m = 2$ as commonly reported in literature), d_{jk} is the Euclidean distance between sample $\mathbf{x}_j$ and center $\mathbf{c}_k$ of the k-th cluster, $\alpha \geq 0$ is a regularization parameter for the second part of the objective function that exploits class information, $b_j = b(\mathbf{x}_j)$ and $f_{jk} = 1$ iff the j-th sample has the same class label of the k-th cluster.

In DISSFCM the SSFCM algorithm is applied incrementally so as to enable continuous update of clusters based on new data chunks. When a new chunk is available, SSFCM granulates data in the chunk by producing a set of K clusters represented by K labeled prototypes $P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_K\}$, representatives for the local data in the current chunk. Each prototype $\mathbf{p}_k$ (with $k = 1, \ldots, K$) is a medoid, i.e. the sample closest to the cluster centroid $\mathbf{c}_k$. Before starting the clustering process, K labeled data are randomly chosen to initialize the prototypes, so that each cluster prototype is associated to a class label. To take into account the evolution of the data during the incremental clustering process, the cluster prototypes discovered from the previous chunk are used to initialize SSFCM for the current chunk.

In order to adapt the model to the evolving structure of data, DISSFCM is equipped with a splitting mechanism [14] that is applied to the current cluster structure in order to split a low-quality cluster into two higher-quality clusters. Denoted by C_k the cluster related to the centroid $\mathbf{c}_k$, its quality is evaluated in terms of the *reconstruction error* [17]

$$V_{\max} = \max_k \sum_{\mathbf{x}_j \in C_k} \|\mathbf{x}_j - \hat{\mathbf{x}}_j\|^2 \quad (2)$$

which measures the difference between the original data $\mathbf{x}_j$ and their reconstructed counterpart $\hat{\mathbf{x}}_j$ that are derived using the cluster prototypes $\mathbf{p}_k$ and membership degrees u_{jk} as follows:

$$\hat{\mathbf{x}}_j = \frac{\sum_{k=1}^{K} u_{jk}^2 \mathbf{p}_k}{\sum_{k=1}^{K} u_{jk}^2} \quad (3)$$

The splitting mechanism is activated when the absolute difference of $V_{\max}$ between two consecutive chunks is above a threshold ϵ. This means that the current number of clusters is not enough to effectively represent the data, hence the number of clusters should be augmented. The cluster having the highest value of the reconstruction error, i.e. the cluster exhibiting the lowest reconstruction ability, is selected as candidate for splitting. The splitting is performed by means of the Conditional Fuzzy Clustering [18] applied to the collection of data samples belonging to the cluster so as to create two novel prototypes. After conditional clustering, the prototype of the split cluster is replaced by two novel prototypes that inherit the class label of the original prototype. Then membership values u_{ik} are recomputed as in SSFCM. The splitting is repeated until the reconstruction error drops below the previous value. A maximum pre-fixed number of splittings is allowed for each chunk, to prevent the creation of too many clusters.

To perform data classification, each sample is matched against all prototypes and assigned to the class label of the best-matching prototype. The matching mechanism is based on the standard Euclidean distance that is assumed to be used in DISSFCM algorithm to group data into clusters. At the end, the algorithm returns the most recent collection of the prototypes, reflecting the data structure of the last data chunk. The returned collection of prototypes can be used as input for a new run of the algorithm as long as new data are available from the data stream.

3 Experimental results

The DISSFCM algorithm was applied on a publicly available dataset[1] of real credit card transactions [19]. Transaction records were gathered over a period of about 48 hours. Every transaction has a timestamp, a monetary amount, and 28 other real-valued, anonymized features. The dataset contains 284,807 transactions, of which 492 were fraudulent. Hence it is highly imbalanced, with the positive class (frauds) accounting for 0.172% of all transactions.

In order to simulate a semi-supervised problem, we assume that data are only partially labeled, hence the fraud detection problem is addressed under three challenging conditions:

1. the number of available labeled examples is small, while the number of unlabeled examples is abundant;

2. the positive examples (fraudulent transactions) are very rare comparing with negative ones (legitimate transactions);

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

	Time slots	#chunks	chunk size
S1	24 hrs	2	142,403
S2	12 hrs	4	71,201
S3	6 hrs	8	35,600
S4	3 hrs	16	17,800
S5	1 hrs	48	5,933

Table 1: Different structures of the data stream considered for experiments.

3. data are not completely available for model building but only in small chunks.

We considered temporal slots to partition the dataset in chunks, in order to simulate a data stream. As reported in Table 1, we considered five different structures of the data stream (denoted by S1, S2, S3, S4, S5) by defining chunks (time slots) of different granularity. Specifically, we considered daily time slots (S1), half-day slots (S2), six hours slots (S3), three hours slots (S4), one hour slots (S5). Table 1 summarizes some statistics on chunks. For each chunk, we considered the subsequent chunk as test set.

Since the credit card dataset is highly imbalanced, prediction could achieve very high accuracy score even without detecting any fraud transaction. Hence proper evaluation metrics for fraud detection should be used [20]. As suggested in [21] we adopted the following metrics to evaluate the predictive ability of the classifier on the target class (fraudulent transaction):

- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 = 2\frac{Recall*Precision}{Recall+Precision}$

High precision denotes low occurrence of false alarms (non-fraudulent transactions that are labelled as fraudulent), while high recall denotes a small risk of overlooking fraudulent transactions. The F1-score gives the harmonic mean of both precision and recall. All these measures are averaged over all the available test sets.

The first set of simulations was devoted to evaluate the accuracy of DISSFCM when changing the chunk granularity (S1, S2, S3, S4, S5) and the percentage of labeled data (25%, 50%, 75%, 100%). The main goal of this study was to observe the influence of the labeling percentage on the classification process.

To show how the classification model produced by DISSFCM evolves during time by dynamically following the data distribution, in fig. 1 we plot the reconstruction error (2) when processing the stream S3 with

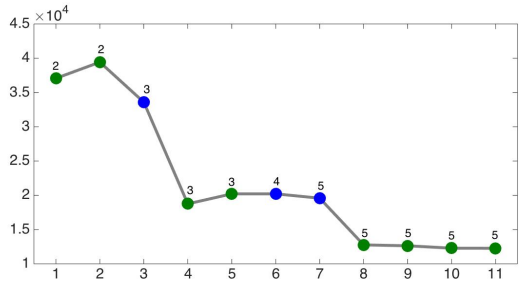

Figure 1: Trend of the reconstruction error in case S3 with %Labeling=25%.

25% of labeling. On the X-axis, we report the evolving stages from 0 to 10, where each stage can be either a data clustering (green dots) or a cluster split (blue dots); on the Y-axis, the reconstruction error (2) is reported. At stage 1, two clusters are generated, one for each class, and the reconstruction error computed. At stage 2, a new chunk is presented and new data are clustered accordingly. Since the reconstruction error has increased, a split is performed and a new cluster is added. This is reported in stage 3, where we observe that the reconstruction error has sensibly decreased. Other splits are applied at stage 6 and 7, leading to a total number of five clusters. From this point on, the reconstruction error stabilizes and the number of clusters remains unchanged.

Tables 2, 3 and 4 show the average precision, recall and F1-score computed on the test sets for DISSFCM by varying the labeling percentage and the chunk granularity. It can be seen that when the chunk granularity decreases (less samples in each chunk), the precision generally increases while the recall decreases. Roughly speaking, as the chunk size becomes small, the classifier becomes more selective (thus reducing false alarms) but it is less sensitive to frauds that follow patterns that are not well represented in small chunks. The F1-score shows that these two phenomena compensate together, and the overall quality of the classifier does not change too much. Depending on the target application, chunk size should be decided so as to foster precision over recall or the contrary.

We observe similar results by varying the labeling percentage. This means that the classification ability of DISSFCM is stable even with limited a priori information on the class of transactions. This is worth in real cases, where only a small portion of the whole data can be labeled by the experts.

In the second set of simulations we compared DISSFCM (with 100% of labeled data) with other supervised incremental learning algorithms. To perform comparison, we considered different methods available

	%Labeling			
	25	**50**	**75**	**100**
S1	0.50(—)	0.55(—)	0.56(—)	0.55(—)
S2	0.59(0.02)	0.60(0.01)	0.61(0.02)	0.61(0.02)
S3	0.75(0.15)	0.75(0.15)	0.75(0.14)	0.75(0.15)
S4	0.76(0.14)	0.72(0.24)	0.72(0.24)	0.67(0.30)
S5	0.73(0.31)	0.69(0.32)	0.71(0.34)	0.73(0.33)

Table 2: Average precision (and standard deviation) on the test sets with DISSFCM, varying the labeling percentage and the chunk granularity.

	%Labeling			
	25	**50**	**75**	**100**
S1	0.79(—)	0.79(—)	0.79(—)	0.79(—)
S2	0.76(0.04)	0.76(0.04)	0.76(0.04)	0.76(0.04)
S3	0.69(0.19)	0.69(0.19)	0.69(0.19)	0.66(0.26)
S4	0.73(0.13)	0.70(0.22)	0.70(0.22)	0.62(0.29)
S5	0.67(0.27)	0.65(0.27)	0.63(0.29)	0.63(0.31)

Table 3: Average recall (and standard deviation) on the test sets with DISSFCM, varying the labeling percentage and the chunk granularity.

in MOA (Massive Online Analysis)[2] [22]. MOA is an open source framework that includes a collection of machine learning algorithms for data stream mining. In particular, we considered seven machine learning algorithms (listed in table 5) including incremental decision tree algorithms (Hoeffding Trees, and its variants Hoeffding Option Trees and Hoeffding Adaptive Tree), an incremental version of Naive Bayes algorithm, and ensemble methods (OCBoost, OzaBag and OzaBoost).

Tables 6, 7 and 8 report the comparative results in terms of average precision, recall and F1-score. It can be seen that there is no method that prevails on the others by varying the chunk granularity. We observe that most methods worsen their performance when the number of data in each chunk decrease. Conversely, DISSFCM provides stable results varying the chunk composition.

As a final remark, we observe that boosted methods achieve the best performance. This is to be expected since they produce ensemble of different classifiers. Despite the good accuracy, ensemble methods produce highly complex classification models that are difficult to interpret. On the contrary, DISSFCM produces simple classification models (with acceptable accuracy) that can be easily interpreted by an expert since they are expressed in form of cluster prototypes that offer a synthetic representation of the data.

[2] `https://moa.cms.waikato.ac.nz`

	%Labeling			
	25	**50**	**75**	**100**
S1	0.62(—)	0.65(—)	0.65(—)	0.65(—)
S2	0.66(0.03)	0.67(0.02)	0.68(0.02)	0.68(0.02)
S3	0.68(0.14)	0.69(0.14)	0.69(0.13)	0.64(0.25)
S4	0.74(0.12)	0.71(0.22)	0.71(0.22)	0.63(0.28)
S5	0.68(0.26)	0.64(0.28)	0.64(0.29)	0.66(0.29)

Table 4: Average F1-score (and standard deviation) on the test sets with DISSFCM, varying the labeling percentage and the chunk granularity.

Model	**Code**
Hoeffding Adaptive Tree [23]	M1
Hoeffding Option Trees [24]	M2
Hoeffding Tree [25]	M3
Naive Bayes incremental learner	M4
OCBoost [26]	M5
OzaBag [27]	M6
OzaBoost [27]	M7
DISSFCM	M8

Table 5: Incremental learning algorithms used for comparison.

4 Conclusions

In this paper we have applied DISSFCM, a dynamic incremental semi-supervised fuzzy clustering to the problem of credit card fraud detection. Preliminary experimental results show that DISSFCM can cope with some challenges characterizing fraud detection: 1) handling the class imbalance, since legitimate transactions far outnumber the fraudulent ones and 2) operating with a small number of recent transactions, that may be partially labeled in the form of investigators feedback. DISSFCM gives comparable results to other incremental methods, and it adds the advantage of dealing with partially labeled data without an significant decay of performance.

Differently from boosted methods, which are more accurate but less interpretable, DISSFCM creates the classification model as a simple collection of cluster prototypes that are easy to read and understand. This is an added value because interpretability plays an important role in the context of credit card fault detection. In fact, automatic tools for FSD are used as decision support systems for human experts that will eventually analyze the activities that have been recognized as fraudulent. Furthermore, experiments show that DISSFCM is still effective with small chunks that correspond to frequent data analysis. This reflect what happens in real scenarios, where a fraudulent activity needs to be detected as soon as it is possible.

	S1	S2	S3	S4	S5
M1	0.12(-)	0.41(0.48)	0.44(0.38)	0.43(0.38)	0.40(0.20)
M2	0.75(-)	0.83(0.04)	0.80(0.04)	0.79(0.08)	0.69(0.13)
M3	0.77(-)	0.82(0.04)	0.79(0.04)	0.79(0.08)	0.69(0.13)
M4	0.98(-)	0.05(0.00)	0.07(0.01)	0.06(0.01)	0.09(0.04)
M5	1.00(-)	0.80(0.07)	0.79(0.03)	0.67(0.14)	0.54(0.11)
M6	1.00(-)	0.92(0.03)	0.88(0.02)	0.87(0.03)	0.76(0.14)
M7	1.00(-)	0.67(0.07)	0.76(0.04)	0.80(0.06)	0.76(0.03)
M8	0.55(-)	0.61(0.02)	0.75(0.15)	0.67(0.30)	0.73(0.33)

Table 6: Comparison in terms of average precision (and standard deviation).

	S1	S2	S3	S4	S5
M1	0.78(-)	0.65(0.09)	0.55(0.08)	0.48(0.15)	0.46(0.10)
M2	0.62(-)	0.50(0.07)	0.54(0.04)	0.53(0.05)	0.41(0.07)
M3	0.62(-)	0.50(0.07)	0.54(0.03)	0.54(0.05)	0.42(0.07)
M4	0.10(-)	0.77(0.05)	0.80(0.02)	0.81(0.03)	0.81(0.02)
M5	0.65(-)	0.69(0.03)	0.66(0.02)	0.61(0.07)	0.47(0.10)
M6	0.74(-)	0.54(0.09)	0.56(0.03)	0.56(0.05)	0.49(0.10)
M7	0.70(-)	0.63(0.06)	0.69(0.04)	0.67(0.04)	0.63(0.06)
M8	0.79(-)	0.76(0.04)	0.66(0.26)	0.62(0.29)	0.63(0.31)

Table 7: Comparison in terms of average recall (and standard deviation).

Future works will be devoted to give better stability of the method, which now shows an appreciable variance in results. This could be achieved by working on more refined ways for transferring the acquired knowledge from one chunk to another. Moreover a merge mechanism is under study, in order to limit the uncontrolled growth of the number of clusters, that could lead the predictive model to overfit the data.

Acknowledgement

The research is partially supported by Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) under grant PON ARS01_01116 "TALIsMAn". The authors are members of the INdAM Research group GNCS.

References

[1] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," *Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017*, vol. 2017-Janua, pp. 1–9, 2017.

[2] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[3] A. Abdullatif, F. Masulli, and S. Rovetta, "Clustering of nonstationary data streams: a survey of fuzzy partitional methods," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1258, 2018.

	S1	S2	S3	S4	S5
M1	0.20(-)	0.38(0.27)	0.39(0.20)	0.36(0.20)	0.39(0.11)
M2	0.68(-)	0.62(0.05)	0.64(0.02)	0.64(0.05)	0.52(0.09)
M3	0.68(-)	0.62(0.05)	0.64(0.02)	0.64(0.05)	0.52(0.09)
M4	0.67(-)	0.10(0.00)	0.12(0.02)	0.12(0.01)	0.16(0.05)
M5	0.82(-)	0.74(0.04)	0.72(0.02)	0.63(0.12)	0.50(0.10)
M6	0.88(-)	0.68(0.07)	0.69(0.02)	0.68(0.04)	0.60(0.11)
M7	0.85(-)	0.65(0.06)	0.72(0.03)	0.73(0.03)	0.69(0.05)
M8	0.65(-)	0.68(0.02)	0.64(0.25)	0.63(0.29)	0.63(0.31)

Table 8: Comparison in terms of average F1-score (and standard deviation).

[4] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review (With Discussion)," *Statistical Science*, vol. 17, no. 3, pp. 235–255, 2002.

[5] 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 Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, 2018.

[6] G. Casalino, C. Castiello, N. D. Buono, and C. Mencar, "A Framework for Intelligent Twitter Data Analysis with Nonnegative Matrix Factorization," *International Journal of Web Information Systems*, vol. 14, no. 3, 2018.

[7] G. Krempl and V. Hofer, "Classification in presence of drift and latency," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 596–603, 2011.

[8] 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. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2010.08.008

[9] R. Bolton and D. J. Hand, "Unsupervised Profiling Methods for Fraud Detection," *Proc. Credit Scoring and Credit Control VII*, vol. VII, pp. 5–7, 2001.

[10] P. Vikrant Agaskar, M. Babariya, S. Chandran, and N. Giri, "Unsupervised Learning for Credit Card fraud detection," *International Research Journal of Engineering and Technology*, pp. 2395–56, 2017. [Online]. Available: https://www.irjet.net/archives/V4/i3/IRJET-V4I3608.pdf

[11] J. T. S. Quah and M. Sriganesh, "Real-time credit card fraud detection using computational intelligence," *Expert systems with applications*, vol. 35, no. 4, pp. 1721–1732, 2008.

[12] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.;

2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, p. 542, 2009.

[13] X. Zhu, "Semi-Supervised Learning Literature Survey," *SciencesNew York*, vol. 10, no. 1530, p. 10, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.99.9681&rep=rep1&type=pdf

[14] G. Casalino, G. Castellano, and C. Mencar, "Incremental adaptive semi-supervised fuzzy clustering for data stream classification." in *Proc. of the 2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS 2018)*, Rhodes, Greece, 5 2018, pp. 1–7.

[15] G. Casalino, G. Castellano, A. Fanelli, and C. Mencar., "Enhancing the dissfcm algorithm for data stream classification." in *Fuzzy Logic and Applications. WILF 2018.*, ser. Lecture Notes in Computer Science. LNAI 10614., M. F. Fullér R., Giove S., Ed. Genova, Italy: Springer, Cham, September 6-7, 2018. 2019, vol. 11291, pp. 109–122, dOI:10.1007/978-3-030-12544-8_9.

[16] G. Castellano and A. M. Fanelli, "Classification of Data Streams by Incremental Semi-supervised Fuzzy Clustering," in *Int. Workshop on Fuzzy Logic and Applications*, ser. Lecture Notes in Computer Science, vol. 10147, 2016, pp. 185–194.

[17] W. Pedrycz, "Algorithms of Fuzzy Clustering with Partial Supervision," *Pattern Recogn. Lett.*, vol. 3, no. 1, pp. 13–20, 1 1985. [Online]. Available: http://dx.doi.org/10.1016/0167-8655(85)90037-6

[18] ——, "Conditional Fuzzy C-Means," *Pattern Recognition Letters*, vol. 17, no. 6, pp. 625–631, 5 1996. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/016786559600027X

[19] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 159–166.

[20] K. Yufeng, L. Chang-Tien, S. Sirwongwattana, and H. Yo-Ping, "Survey of fraud detection techniques," *IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 2, pp. 749–754, 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1297040/

[21] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, pp. 1–21, 03 2015.

[22] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "{MOA:} Massive Online Analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?id=1859903

[23] A. Bifet and R. Gavaldà, "Adaptive Learning from Evolving Data Streams," in *Advances in Intelligent Data Analysis VIII*, N. M. Adams, C. Robardet, A. Siebes, and J.-F. Boulicaut, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 249–260.

[24] B. Pfahringer, G. Holmes, and R. Kirkby, "New Options for Hoeffding Trees," in *AI 2007: Advances in Artificial Intelligence*, M. A. Orgun and J. Thornton, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 90–99.

[25] G. Hulten, L. Spencer, and P. Domingos, "Mining Time-changing Data Streams," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. ACM, 2001, pp. 97–106. [Online]. Available: http://doi.acm.org/10.1145/502512.502529

[26] R. Pelossof, M. Jones, I. Vovsha, and C. Rudin, "Online coordinate boosting," *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 1354–1361, 2009.

[27] N. Oza, "Online Bagging and Boosting," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 2019, pp. 2340–2345. [Online]. Available: http://ieeexplore.ieee.org/document/1571498/