Advances in Data Analysis and Classification
https://doi.org/10.1007/s11634-022-00515-5

REGULAR ARTICLE

The role of diversity and ensemble learning in credit card fraud detection

Gian Marco Paldino[1] · Bertrand Lebichot[1] · Yann-Aël Le Borgne[1] · Wissam Siblini[3] · Frédéric Oblé[3] · Giacomo Boracchi[2] · Gianluca Bontempi[1]

Received: 31 January 2021 / Revised: 18 July 2022 / Accepted: 8 August 2022

Abstract

The number of daily credit card transactions is inexorably growing: the e-commerce market expansion and the recent constraints for the Covid-19 pandemic have significantly increased the use of electronic payments. The ability to precisely detect fraudulent transactions is increasingly important, and machine learning models are now a key component of the detection process. Standard machine learning techniques are widely employed, but inadequate for the evolving nature of customers behavior entailing continuous changes in the underlying data distribution. his problem is often tackled by discarding past knowledge, despite its potential relevance in the case of recurrent concepts. Appropriate exploitation of historical knowledge is necessary: we propose a learning strategy that relies on diversity-based ensemble learning and allows to preserve past concepts and reuse them for a faster adaptation to changes. In our experiments, we adopt several state-of-the-art diversity measures and we perform comparisons with various other learning approaches. We assess the effectiveness of our proposed learning strategy on extracts of two real datasets from two European countries, containing more than 30 M and 50 M transactions, provided by our industrial partner, Worldline, a leading company in the field.

Keywords Finance · Fraud detection · Concept drift · Ensemble learning · Diversity

Mathematics Subject Classification 68T05 Learning and adaptive systems in artificial intelligence

✉ Gian Marco Paldino
gpaldino@ulb.ac.be

[1] Machine Learning Group, Computer Science Departement, Faculty of Sciences, Université Libre de Bruxelles, Bruxelles, Belgium

[2] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

[3] Research, Development and Innovation, Worldline, Lyon, France

Published online: 28 September 2022

1 Introduction

The use of credit cards is constantly growing. This is fertile ground for fraudsters, with new technologies providing new methods to carry out frauds. For instance, the recent COVID-19 pandemic permanently changed customers habits (Accenture 2020) favoring electronic payments. Prompt detection of fraudulent behavior does not only prevent an economic loss, but also preserves clients' trust in the company. The enormous amount of transactions provides profitable data, and the usage of machine learning for fraud detection is crucial to extract hidden knowledge and improve fraud detection accuracy. Standard machine learning algorithms are effective only when their requisites are met (Batista et al. 2000). The fraud detection problem is challenging for its characteristic of not meeting several of the standard learning algorithms requisites: data is heavily *unbalanced* (Dal Pozzolo et al. 2013) (i.e. genuine transactions outnumber frauds) and subject to *concept drift* (Gama et al. 2004) (i.e. data distribution changes over time); the two classes might *overlap* (i.e. no clear fraud/genuine distinction) and customers might report frauds several days after their perpetuation, leading to a *verification latency* (Dal Pozzolo et al. 2017) (i.e. delay in availability of labeled transactions).

Additionally, it is possible that recurrent concepts appear in data: customers might have seasonal behavior, or the market might possibly follow a periodic pattern. We propose a learning strategy that takes advantage of possible recurrent concepts, without losing in terms of adaptability to changes in the underlying distribution. This is achieved by using an *ensemble* (Zhou 2009) as an historical memory of models, making several past concepts ready-to-use in case of re-appearance. To achieve that, we employ *Diversity* criteria (Kuncheva 2004; Sun et al. 2018) to choose which models need to be preserved. We also propose a weighting strategy specifically thought for the fraud detection problem.

Our study aims at answering the following questions:

- Does the use of diversity criteria improve the performance of an ensemble-based transactions classifier?
- How do different diversity measures impact the final performance?
- Is a diversity-based ensemble effective in a real scenario with possible concept drift?

The main contributions of this paper are:

1. We assess the impact of diversity-based and time-based ensemble techniques in fraud detection
2. We compare five state-of-the-art diversity measures
3. We analyze ensemble-based and non-ensemble-based approaches
4. We propose a weighted diversity-based ensemble learning strategy

This study can be considered complementary to our previous work on diversity and transfer learning (Lebichot et al. 2020), whose focus was on Neural Networks (NN) and Deep Neural Networks (DNN). Here we consider exclusively Random Forests (RF) for the following reasons: (1) they have shown to overperform other models in previous studies on fraud detection (Dal Pozzolo 2015) (2) they are beneficial to establish feature importance, which is highly valued by investigators to understand the causes behind labeling a transaction (3) they provide a natural ensemble that can be easily exploited to tackle problems such as imbalance, by providing each decision tree with a balanced subset of the original data (explained in Sect. 5.3).

This paper is organized as follows: Sect. 2 introduces the context of our study to give the necessary background for understanding the problem formulation in Sect. 3. Our proposed learning strategy can be found in Sect. 4 while Sect. 5 is dedicated to its experimental assessment. The conclusions of this work can be found in Sect. 6.

2 Background and related work

In this section we outline all the required knowledge to properly contextualize the problem of credit card fraud detection. We start by briefly describing the structure of a real-life fraud detection system in Sect. 2.1, inspired by the one used by our industrial partner. We then present in Sect. 2.2 a brief outline of how the problem of fraud detection has been studied in the literature. In Sect. 2.3 we focus on credit card transactions data and we illustrate its challenges: unbalancedness, streaming nature, and variability. Finally, we introduce some methods used to tackle the above issues, by focusing in Sect. 2.4 on the concept of Diversity, main components of our *Proposed Learning Strategy* (Sect. 4).

2.1 Fraud detection system

A real-life Fraud Detection System typically employs five levels of control: (1) the *terminal*, (2) the *transaction blocking rules*, (3) the *scoring rules*, (4) the *data driven model* and (5) the *investigators*. (1) The *terminal* performs standard security checks on all transactions requests, as the correctness of the PIN code, the card status, current balance, etc. (Van Vlasselaer et al. 2015). (2) The *transaction blocking rules* are confidential conditional statements to stop clear fraudulent transactions (e.g. "IF blacklisted website THEN deny transaction"). (3) The *scoring rules* are confidential conditional statements to score potential fraudulent transactions (e.g. "IF amount >> average THEN fraud score = 0.8"). (4) The *data driven model* adopts a statistical model, trained on past labeled data, to estimate the probability of each transaction being a fraud. This is the layer that this works aims at improving. (5) The *investigators* are experienced professionals that design layers (2) and (3) rules. Additionally, they check the riskiest transactions from level (4) to return a true label for each of them: the former are referred as *alerts* and the latter as *feedbacks*. An extended description of a *Fraud Detection System* can be found in (Dal Pozzolo et al. 2017).

2.2 Fraud detection literature

Despite the limitations related to the availability of data for privacy reasons, credit card fraud detection has always interested machine learning literature for its challenging nature and its relevant impact. Supervised learning methods (Brause et al. 1999; Chan et al. 1999) and unsupervised methods (Bolton et al. 2001; Phua et al. 2010) have been proposed, with the former gaining more popularity than the latter. In supervised learning, fraud detection is achieved by training a classifier on labeled transactions and using it to classify authorized transactions. Several classification algorithms have been used in fraud detection, from Logistic Regression (Jha et al. 2012) to Support Vector Machines (Whitrow et al. 2009), from Decision Trees (Dal Pozzolo et al. 2014b) to Neural Networks (Dorronsoro et al. 1997). Random Forest achieved the best performance in different cases (Bhattacharyya et al. 2011; Dal Pozzolo et al. 2015a; Dal Pozzolo et al. 2014a). Unsupervised learning approaches the problem by detecting transactions that differ from the majority, for instance by means of Peer Group Analysis (Weston et al. 2008), and other clustering algorithms (Phua et al. 2010). Combinations of supervised and unsupervised learning have also been proposed (Carcillo et al. 2019; Veeramachaneni et al. 2016). Strategies have been also proposed to re-use already existing knowledge by means of *domain adaptation* techniques (Lebichot et al. 2019).

The statistical literature also addresses fraud detection (Bolton and Hand 2002). From traditional statistical methods such as linear discriminant analysis (Hand 1981) to more complex models such as Neural Networks (Webb 2003); from rule-based methods (Clark and Niblett 1989) to tree-based algorithms (Breiman et al. 2017), statistics has always been the core of the fraud detection literature. Other statistical techniques often considered include, for instance, outlier detection (Hodge and Austin 2004), sampling (Domingo et al. 2002) and graph mining (Washio and Motoda 2003). A comprehensive survey can be found here (Phua et al. 2010).

More recently, authors in Cerioli et al. (2018) propose a Fraud Detection System based on the Newcomb–Benford law for significant digits, trying to establish conditions for the validity of the law in the field of international trade data. Rousseeuw et al. (2019) propose a Robust Time Series Monitoring approach based on outlier-detection, and propose a double wedge plot as an effective visualization tool. Cerasa and Cerioli (2017) study the problem of merging homogeneous groups of pre-classified observations from a robust perspective motivated by the anti-fraud analysis of international trade data, running simulations under different contamination scenarios. Graph-based approaches have also shown good potential in the fraud detection literature: a systematic literature review can be found in Pourhabibi et al. (2020).

2.3 Specific challenges of fraud detection

Transactions data is characterized by several aspects that make the fraud detection problem challenging, the most important are: the *data unbalancedness*, the *delayed labels availability*, and its *evolving nature*. A detailed description of the challenges in the fraud detection problem can be found in Alazizi et al. (2019), where the authors assess their results on data from our industrial partner.

Fraud detection data is heavily *unbalanced*, meaning that the number of frauds is exceptionally small compared to the genuine transactions. Formally, in a binary classification task $f : \mathbb{R}^n \rightarrow \{0, 1\}$, with input $\mathrm{X} \in \mathbb{R}^n$ and output $\mathrm{Y} \in \{0, 1\}$, a training set of size N can be defined as follows:

$$T_N = \{(x_1, y_1), \ldots, (x_N, y_N)\} \tag{1}$$

and it is *unbalanced* when the number of positive cases N^+ is small (resp. big) compared to the number of negative cases N^-. Notice that $N^+ + N^- = N$.

Specifically, the number of frauds can be less than 1 transaction per 1000 (Juszczak et al. 2008), making the distribution highly skewed towards the majority class. Most machine learning algorithms are not meant to work with an unbalanced dataset (Batista et al. 2000): their tendency is in fact to favor, in predictions, the most frequent class. Standard methods for dealing with unbalanced data are *Undersampling* (He and Garcia 2009) and *Oversampling* (Drummond and Holte 2003). They respectively reduce the majority class or size up the minority class to compensate for the imbalance. Several variations exist, here we name SMOTE (Chawla et al. 2002), ADASYN (Haibo et al. 2016), and EasyEnsemble (Liu et al. 2008). Recent literature (Ba 2019; Mullick et al. 2019) has shown the usage of Generative Adversarial Networks to create realistic frauds for rebalancing the two classes.

When a fraud occurs, it might take several days for a cardholder to realize it and dispute the transaction. Hence, it is unrealistic to know the true nature of a fraud without considering a *verification latency*. A small fraction of transactions receives its true label: feedbacks provided by the investigators over the riskiest transactions represent the most recent supervised information available to the Fraud Detection System. The *verification latency* is a serious drawback in data availability, being recent data the most representative of the current data distribution. This issue is often not considered in the literature (e.g. (Bolton et al. 2001; Brause et al. 1999)) and tests are performed with the real labels available one day after the transaction took place. Recent works (Dal Pozzolo 2015; Dal Pozzolo et al. 2017; Lebichot et al. 2016) have demonstrated the importance of considering the set of feedbacks and the set of available transactions separately. This is presented formally in Sect. 3.

A key assumption in the adoption of standard classification algorithms is that the underlying distribution of data does not change over time. When the learning task occurs in an *evolving* environment, this assumption is typically not met: this problem is defined as *Dataset Shift*, (Quionero-Candela et al. 2009) or *Concept Drift* (Gama et al. 2014a; Widmer and Kubat 1996; Schlimmer and Granger 1986). The goal of a traditional classification problem is to estimate $\mathcal{P}(y \mid x)$, that can be written, from Bayes law, as follows:

$$\mathcal{P}(y \mid x) = \frac{\mathcal{P}(x \mid y)\mathcal{P}(y)}{\mathcal{P}(x)} \tag{2}$$

Concept Drift might hence be formally defined as $\mathcal{P}_t(x, y) \neq \mathcal{P}_{t+1}(x, y)$ and the joint distribution of a sample (x,y) can be written as follows:

$$\mathcal{P}(x, y) = \mathcal{P}(y \mid x)\mathcal{P}(x) = \mathcal{P}(x \mid y)\mathcal{P}(y) \tag{3}$$

From (2) and (3) we can identify its possible sources: it can be caused by a change in $\mathcal{P}(y \mid x)$, $\mathcal{P}(x \mid y)$, or $\mathcal{P}(y)$. Notice that a change in $\mathcal{P}(x)$ does not affect y and is not relevant. If $\mathcal{P}_t(y) \neq \mathcal{P}_{t+1}(y)$, a miscalibration of classifiers can happen, for instance if the number of frauds suddenly increases. If $\mathcal{P}_t(x \mid y) \neq \mathcal{P}_{t+1}(x \mid y)$, the distribution within the classes changes, but the class boundary remains the same. An example can be the discover of a new fraud technique. If $\mathcal{P}_t(y \mid x) \neq \mathcal{P}_{t+1}(y \mid x)$, the class boundary changes, leading to biased classifiers, e.g. with the change of customer habits. The problem has been widely studied by the literature (Gama et al. 2004, 2014b; Holte et al. 1989), and the proposed approaches can be classified in two broad categories (Alippi et al. 2013): Active Adaptation (Gama et al. 2004) and Passive Adaptation (Žliobaite 2010). Identifying the presence of Concept Drift, for instance by a drop in performance, is the key component of Active Adaptation methods, while Passive Adaptation relies on a continuous update of the model, regardless of whether Concept Drift is present or not, favoring more recent data. When faced with Concept Drift adaptation, the literature often deals with the problem of *Catastrophic Forgetting* (Kirkpatrick et al. 2017), which is the abrupt loss of previously learned information upon learning new information. However, the problem of catastrophic forgetting is more related to Neural Network based learners and is not treated in this work, focusing on Decision Tree-based learners.

2.4 Diversity

Working with an ensemble of models might have different disadvantages, based on how the individuals are managed. Preserving several models is expensive in terms of memory and computation and if the criteria to select which model to store is based on *recentness*, we might end up with needless redundancy. Moreover, when a concept from the past would reappear in the future, none of the stored models would be suitable, for their training-set being only originated from recent data. Promoting *diversity* to select which model to store would avoid having redundant knowledge and allow flexibility in case of recurrence of past knowledge. Diversity among an ensemble of classifiers is considered a key component in the learning process (Cunningham and Carney 2000). However, no general formal definition of diversity exist. Several measures of similarity among binary classifiers have been used in the literature to compute diversity (see Sect. 4.1). A study about the impact of diversity measures on the accuracy of an ensemble of classifiers can be found in (Kuncheva 2004). A recent interesting approach has proposed a novel framework to tackle concept drift by means of diversity: DTEL (Sun et al. 2018) proposes to maintain a fixed buffer and store models that maximize a diversity measure, and additionally to adapt members of the ensemble to recent data by means of *transfer learning*.

3 Problem formulation

In order to clearly understand the proposed approach and the development of the experiments, a clear formulation of the fraud problem is necessary. It is important to emphasize the simplifying nature of the following problem formulation: real-life Fraud Detection Systems are complex and rich in details, we extract the essential aspects for the learning procedure.

The ith authorized transaction is identified by a couple (x_i, y_i), where $x_i \in R^n$ is the feature vector (eg. *transaction amount, date, etc*) and $y_i \in \{0, 1\}$ represents its class, 0 standing for *genuine* and 1 for *fraudulent*. The *streaming nature* of data is simplified by considering new data arriving once a day in *chunks* or *batches*. We denote the batch arrived on day t by B_t. The size of the batch $s = |B_t|$ denotes the number of transactions in a specific batch. We can define a batch as follows

$$\mathrm{B}_t = \left\{(x_j, y_j), (x_{j+1}, y_{j+1})..., (x_{j+s-1}, y_{j+s-1})\right\} \tag{4}$$

A classifier $\mathcal{K}_{t|\delta}$ is created when a new *batch* of data becomes available. $\mathcal{K}_{t|\delta}$ is trained with the δ most recent *batches* available at day t. For instance, $\mathcal{K}_{10|3}$ will use for its training the most recent 3 chunks of data available at day 10, respectively $\mathrm{B}_7, \mathrm{B}_8, \mathrm{B}_9$. Equivalently we can use the following notation:

$$\mathcal{K}_{t|\delta} = \text{TRAIN}\left(\{\mathrm{B}_{t-1}, \ldots, \mathrm{B}_{t-\delta}\}\right) \tag{5}$$

The fraud detection problems is hence formulated as a binary classification problem where a classifier $\mathcal{K}_{t|\delta}$ associates to each features vector $x_i \in R^n$ a label $y_i \in \{0, 1\}$. We denote the probability for x_i to be a fraud according to $\mathcal{K}_{t|\delta}$, also known as the *posterior* of $\mathcal{K}_{t|\delta}$ with the following:

$$\mathcal{P}_{\mathcal{K}_{t|\delta}}\left(1 \mid x_i\right) \tag{6}$$

Furthermore, after a classifier $\mathcal{K}_{t|\delta}$ has processed $s = |B_t|$ transactions from batch B_t, it produces s values of $\mathcal{P}_{\mathcal{K}_{t|\delta}}\left(1 \mid x_i\right)$ that can be ranked according to their *risk* of being fraud, $r\left(x_i\right) \in \{1, \ldots, s\}$. The transaction with highest probability of being a fraud will rank first, and the last one will rank s^{th}. Thereby, the risk function r maps each transaction to its ranking. We extract the *k-most* risky transactions—called *alerts*—as follows:

$$\mathrm{A}_t = \{x_i \in \mathrm{B}_t \mid r\left(x_i\right) \leq k\} \tag{7}$$

It is important to remark that real life risk-functions might not be limited to the probability of a transaction being a fraud: e.g. if the amount of the concerned transaction is particularly low, the risk could be reduced, even with a high probability of being a fraud.

In real life Fraud Detection Systems, because of the time and cost constraints of verification, the set A_t is considered the only set of transactions that can be sent to investigators to check. Investigators provide *feedbacks* about the cards involved in the A_t transactions. A *feedback* is a couple (x_i, y_i) providing the correct classification label back to the Fraud Detection Systems. The number of *feedbacks* can be larger than the size of A_t, because if a transaction is fraudulent, all the transactions belonging to the same card—even outside of A_t—are typically considered fraudulent. The *feedbacks* set can be modeled as follows:

$$\mathrm{F}_t = \{(x_i, y_i) \mid x_i \in \text{cards}\left(\mathrm{A}_t\right)\} \tag{8}$$

where cards (A_t) denotes the set of cards having one or more transactions in A_t. F_t is the only source of correct classification labels for recent data. In fact, we can assume that a η-days period of time called *verification latency* has to pass before knowing true nature of a transaction. Customers could report frauds several days after they took place and and all unreported transactions are considered genuine only after η days, with η often considered constant to simplify computations. This implies that between day t and $t-\eta$ labels are not available, except from transactions contained in F_t. Despite their reduced size w.r.t. the original batch size, *feedbacks* play an important role in the prediction accuracy being the most recent data available.

4 Proposed learning strategy

We start by describing the necessary components of our proposed learning strategy, namely an *ensemble* of classifiers, the effect of considering a *verification latency*, and the criterion for selecting the ensemble members.

A classifier can be used directly to process authorized transactions from consequent days $(t+1)$, $(t+2)$, ..., or can be used together with other classifiers in an *Ensemble*. An *ensemble* of size n is a collection of n classifiers and is identified by the following notation:

$$\mathcal{E} = \left\{\mathcal{K}_{t_1|\delta_1}, \mathcal{K}_{t_2|\delta_2}, ..., \mathcal{K}_{t_n|\delta_n}\right\} \tag{9}$$

If an *ensemble* $\mathcal{E}$ is used for prediction, the probability of x_i to be a fraud described in (6) becomes:

$$\mathcal{P}_{\mathcal{E}}\left(1 \mid x_i\right) = \left\{\frac{w_0\mathcal{P}_{\mathcal{K}_{t_1|\delta_1}}\left(1 \mid x_i\right) + w_1\mathcal{P}_{\mathcal{K}_{t_2|\delta_2}}\left(1 \mid x_i\right) + ... + w_n\mathcal{P}_{\mathcal{K}_{t_n|\delta_n}}\left(1 \mid x_i\right)}{\sum_i w_i}\right\} \tag{10}$$

where $\{w_0, w_1, ..., w_n\}$ are the *weights* associated with each classifier in the *ensemble*.

Considering the *verification latency* η and the *feedbacks* F described in (8), the literature (Dal Pozzolo 2015; Dal Pozzolo et al. 2017) suggests the importance to treat those labeled transactions separately, adding another layer of complexity to the detection problem. Despite the importance of feedback samples, our work aims only at assessing the impact of diversity, hence we ignore them. However, in order not to stray from reality, we consider the most recent batches of data as a *gap set* that is not available (Fig. 1). We can conclude that in reality a classifier $\mathcal{K}_{t|\delta}$ can only access fully labeled batches prior to day $t-\eta$:

$$\mathcal{K}_{t|\delta} = \mathrm{TRAIN}\left(\left\{\mathrm{B}_{t-\eta-1}\ldots, \mathrm{B}_{t-\eta-\delta}\right\}\right) \tag{11}$$

It is important to remark that an ensemble $\mathcal{E}$ can contain classifiers trained using completely different sets of batches. In particular, the *ensemble* members selection is a key point of an effective learning procedure. Generally, to get a *good ensemble*, its members should *disagree* on some input (Krogh and Vedelsby 1995). We can assume

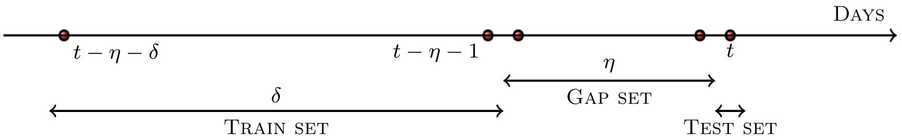

Fig. 1 Real-life FDS scenario, with current day at t. Between day $t-1$ and $t-\eta$ only *feedbacks* from investigators are available: they require specific treatment (Dal Pozzolo 2015) and are ignored here. Hence our Train set (of size δ) covers days $[t-\eta-\delta, t-\eta-1]$; the Gap set (of size η) days $[t-\eta$ to $t-1]$ and our Test set (of size 1) day $[t]$

that from a set $\mathcal{C}$ of c classifiers it is possible to *select* the members for the ensemble $\mathcal{E}$ of s classifiers (with $s < c$), by means of a criterion that leads to a *good ensemble*. We denote by SEL() the function that performs this selection by choosing between all the possible s-sized subsets of $\mathcal{C}$. Being $\mathcal{E} = \text{SEL}(\mathcal{C})$, we have:

$$\mathcal{E} = \text{SEL}(\{\mathcal{K}_{t_1|\delta_1}, \ldots, \mathcal{K}_{t_s|\delta_s}, \ldots, \mathcal{K}_{t_c|\delta_c}\}) = (\left\{\mathcal{K}_{\bar{t_1}|\bar{\delta_1}}, \ldots, \mathcal{K}_{\bar{t_s}|\bar{\delta_s}}\right\}) \qquad (12)$$

The following focuses on how to craft an optimal selection criteria $\text{SEL}(\mathcal{C})$. To address Concept Drift, it is crucial to keep the classifiers continuously updated, by including the most recent knowledge and discarding past information. This can be achieved in two strategies:

- by using a single classifier trained on most recent data (I)
- by using an ensemble of recent classifiers (II)

The aforementioned strategies focus on discarding past information. However, historical knowledge is not always obsolete. Periodic patterns tend to appear in customers' behavior (e.g. during Christmas holidays). *Recurring concepts* suggest the need to exploit historical knowledge rather than discarding it. The exploitation must be done carefully: past data can be beneficial in the presence of recurring concepts but detrimental in its absence. For example, if the distribution of data has changed over time, keeping past data in the train-set of a classifier negatively affects its performance. A variation of strategy (II) can be formulated by using the ensemble as a memory of *relevant* knowledge rather than *recent* knowledge. In particular, the presence of several different concepts, despite their distance in time, could lead to an easier adaptation to their recurrence in the future. Additionally, recent classifiers can be similar and their ensemble redundant. This suggests the use of a *Diversity* measure to choose the candidates classifiers for the ensemble: similar classifiers are discarded, leaving room for diverse ones, that potentially encompass different concepts. Each classifier is weighed based on its most recent performance, using fraud-detection specific metrics.

Our proposed learning strategy, detailed in Algorithm 1, implements the proposed variation of strategy (II). A new candidate classifier $\mathcal{K}_{t|\delta}$ is created every day and trained on the most recent δ labeled batches available. $\mathcal{K}_{t|\delta}$ is then included in an ensemble $\mathcal{E}$ *only* if the *diversity* in the ensemble increases by replacing one of its members with $\mathcal{K}_{t|\delta}$. In this way, we benefit from the *most diverse ensemble* at each day, which is then used to process new transactions by means of weighted average prediction of its members. With a new classifier trained every day and a diversity criterion, we aim at not keeping redundant knowledge, to ideally have an ensemble that stores *diverse past concepts*.

The rationale behind the proposed learning strategy is to seek balance in the *stability-plasticity dilemma* (Mermillod et al. 2013): we would like our classifier to be highly flexible and adaptable to changes, but at the same time, we do not want it to be easily affected by noise or short term changes. Additionally, it is reasonable to assume the existence of recurrent patterns in transactions data, suggesting the importance of not discarding relevant past knowledge.

Algorithm 1 Proposed Learning Strategy

```
procedure
    $t \leftarrow t_0$
    while true do
        for each classifier $\mathcal{K}_i \in \mathcal{E}_{t-1}$ do
            $\mathcal{P}_t[i] \leftarrow$ weighted prediction using $\mathcal{K}_i$                ▷ prediction

        $\mathcal{K}_{t|\delta} \leftarrow$ new classifier, TRAIN $(\{\mathrm{B}_{t-\eta-1}, \ldots, \mathrm{B}_{t-\eta-\delta}\})$                ▷ training

        if $\mathcal{E}_{t-1}$ is not full then                ▷ ensemble update
            $\mathcal{E}_t \leftarrow \mathcal{E}_{t-1} \bigcup \{\mathcal{K}_{t|\delta}\}$
        else
            $\mathcal{E}'_t \leftarrow \mathcal{E}_{t-1} \bigcup \{\mathcal{K}_{t|\delta}\}$
            $\overline{\mathcal{K}}_{t|\delta} \leftarrow \mathrm{argmax}_{\mathcal{K}}$ Div $(\mathcal{E}'_t \setminus \{\mathcal{K}\})$                ▷ diversity computation
            $\mathcal{E}_t \leftarrow \mathcal{E}'_t - \{\overline{\mathcal{K}}_{t|\delta}\}$

        for each classifier $\mathcal{K}_i \in \mathcal{E}_t$ do
            $W_t[i] \leftarrow$ compute weights for $\mathcal{K}_i$                ▷ weights computation

        $t \leftarrow t+1$
```

4.1 Implementation of the proposed learning strategy

As mentioned in Sect. 3, an accurate weighing of the ensemble members is crucial to avoid past knowledge to drastically reduce the accuracy of the final prediction. Given a performance metric $\mathcal{M}_t \in [0, 1]$ computed over the last labeled batch B_t available, we implement the weight of each classifier $\mathcal{K}_i$ for prediction at time step $t + \eta + 1$ as follows:

$$\mathrm{weight}_t^i = \frac{1}{(1 - \mathcal{M}_t) + \epsilon} \tag{13}$$

where ϵ is a small positive value to prevent the denominator from being 0. We set $\epsilon = 0.1$, which implies that a good classifier ($\mathcal{M}_t \sim 1$) has $\mathrm{weight}_t^i \sim 10$ and a bad classifier ($\mathcal{M}_t \sim 0$), has $\mathrm{weight}_t^i \sim 1$.

We compute diversity starting from five pairwise similarity and dissimilarity measures between classifiers: Yule's Q statistic (Yule 1900), Correlation Coefficient

(Benesty et al. 2009), Disagreement measure (Ho 1998), Double Fault measure (Giacinto and Roli 2001), Interrated Agreement measure (Fleiss et al. 1981). The choice of those criteria is motivated by their popularity as similarity measures in the literature (Kuncheva 2004). Starting from a similarity measures $\mathcal{S}(\mathcal{K}_i, \mathcal{K}_j)$ between two classifiers $\mathcal{K}_i$ and $\mathcal{K}_j$, diversity of the ensemble $\mathcal{E}$ at time t is measured as follows:

$$\operatorname{div}(\mathcal{E}_t) = 1 - \frac{1}{\sum_{1 \le i \ne j \le s} 1} \sum_{1 \le i \ne j \le s} |\mathcal{S}(\mathcal{K}_i, \mathcal{K}_j)| \tag{14}$$

where $s = |\mathcal{E}_t|$ is the size of the ensemble. The weighting mechanism in (13) and the computation of diversity as in (14) are inspired from (Sun et al. 2018), but our work diverges from their idea for the following reasons: (a) they adopt only one diversity measure, the Yule's Q statistics (15) (Yule 1900), because it is one of the most popular diversity measures in the literature, with no additional justification; (b) they use a weighting metric based on the Mean Squared Error (MSE), while our weighting metric is specific for fraud detection, because it employs Precision Top-K (see Sect. 5.1) which is often adopted in this case study; and (c) they include directly a *transfer learning* module, without assessing the impact of the diversity ensemble exclusively.

In this work we consider multiple diversity measures, multiple-batches train-set and other major differences due to the fraud detection framework (eg. the existance of a *gap-set*). The mentioned diversity criteria are defined as follows, where N^{ab} is the number of examples for which the classification result is a by f_i and b by f_j, 1 represents a correct classification and 0 represents a misclassification:

- Yule's Q statistic (Yule 1900):

$$Q\left(f_i, f_j\right) = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{15}$$

  The intuition behind Yule's Q is that it computes the ratio between the *difference* and the *sum* of concordant and discordant classifications. If there is only discordant pairs, the value is going to be -1, implying strong disagreement (perfect negative correlation). If there is only concordant pairs the value is going to be +1, meaning perfect positive correlation. If the numbers of agreements and disagreements coincide, the statistic is going to be 0, implying that the two classifiers are not associated.

- Correlation coefficient (Benesty et al. 2009):

$$\rho_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{\left(N^{11} + N^{10}\right)\left(N^{01} + N^{00}\right)\left(N^{11} + N^{01}\right)\left(N^{10} + N^{00}\right)}} \tag{16}$$

  The meaning of the Correlation Coefficient is very similar to the Yule's Q statistics one. It can be proven, in fact, that they have the same sign and that $|\rho| \le |Q|$. The only difference is the denominator, that multiplies the four following scenarios:

number of times the first classifier is correct, no matter the second one; number of times the first classifier is incorrect, no matter the second one, number of times the second classifier is correct, no matter the first one; and number of time the second classifier is incorrect, no matter the first one.

- Disagreement measure (Ho 1998):

$$Dis_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{17}$$

As the name suggests, the disagreement measure represents the ratio of *disagreements* over the total number of observations. For clarity, the disagreement is when one classifier is correct and the other classifier is wrong on one observation, or viceversa. This ratio will get to 1 if the two classifier disagree on all possible observations and to 0 if they agree on all of them.

- Double Fault measure (Giacinto and Roli 2001):

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{18}$$

Again a ratio over all possible observations, but this time only of the observations that were wrongly predicted by both the classifiers. This value goes to 0 if all the observations are correctly classified by at least one of the two classifiers. It goes to 1 if all the observations are wrongly classified by both at the same time.

- Interrated Agreement measure (Fleiss et al. 1981):

$$\kappa_p = \frac{2\left(N^{11}N^{00} - N^{01}N^{10}\right)}{\left(N^{11} + N^{10}\right)\left(N^{01} + N^{00}\right) + \left(N^{11} + N^{01}\right)\left(N^{10} + N^{00}\right)} \tag{19}$$

This measure an additional variation of the Yule’s Q and the Correlation coefficient. More weight is given to the numerator, while the denominator sums the pairwise product of the four scenarios mentioned for the Correlation coefficient.

All previous measures decrease when diversity increases, except for the Disagreement measure, for which (14) becomes:

$$\operatorname{div}(\mathcal{E}_t) = \frac{1}{\sum_{1\leq i\neq j\leq s} 1} \sum_{1\leq i\neq j\leq s} Dis_{i,k} \tag{20}$$

5 Experiments

Our experiments are organized as follows: we first describe the metrics used to assess the performance of a transactions classifier in Sect. 5.1; followed by a description of our datasets in Sect. 5.2. Then, in Sect. 5.5 we outline our experimental settings and we present our results in Sect. 5.6. A discussion about the obtained results is held in Sect. 5.7.

5.1 Metrics

In order to assess the performance of a classifier $\mathcal{K}^{\eta}_{t|\delta}$ (or an *ensemble* $\mathcal{E}$), we adopt the two metrics Precision and Recall, defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{21}$$

where TP, or *True Positive*, indicates the number of correctly classified frauds; FP, or *False Positive*, indicates the number of wrongly classified frauds and FN, or *False Negative*, indicates the wrongly classified genuine transactions. Furthermore, a common metric in the fraud detection literature is the *Precision Top k* (P_k). It measures the ability of the classifier to correctly detect the k-most risky transactions:

$$\text{P}_k = \frac{\text{TP}_k}{k} \tag{22}$$

where TP_k indicates the number of True Positive in the k-most risky transactions (higher predicted probability of being a fraud). The aforementioned metrics depend on the classification threshold $\gamma > 0$, since a transaction is classified as fraudulent if $\mathcal{P}_{\mathcal{K}_{t|\delta}}(x_i) > \gamma$. In our case, a transaction is classified as fraudulent if its probability of belonging to the fraud class is greater than 0.50. Thereby, the value of γ considered in the following experiments for the P_k is $\gamma = 0.5$. Another popular measure that takes into account all possible threshold values is the *Area Under the Precision Recall Curve* (`prauc`): the values of Precision and Recall are plotted for each possible $\gamma \in [0, 1]$ and the area under the resulting curve is measured. Another interesting technique is presented in (Siblini et al. 2020): the calibration of standard metrics. In fact, the number of fraud to detect varies for each test day, affecting metrics as the `prauc`. A *Calibrated Area Under the Precision Recall Curve* (`cprauc`), invariant to the fraud prior π, can be obtained by fixing a reference ratio π_0 and weighing the count of TP and FP in order to calibrate them to the value that they would have if π was equal to π_0, as follows:

$$\text{Prec}_c = \frac{\text{TP}}{\text{TP} + \frac{\pi(1-\pi_0)}{\pi_0(1-\pi)}\text{FP}} = \frac{1}{1 + \frac{\pi(1-\pi_0)}{\pi_0(1-\pi)}\frac{\text{FP}}{\text{TP}}} \tag{23}$$

Once a fraud is discovered, the corresponding credit card is blocked, hence considering multiple transactions from the same card in the metrics measurement can lead to an overestimation of performance. *Grouping by Card*, considering only the highest fraud score between different transactions from the same card, is a good practice.

5.2 Our datasets

In the following, we use extracts from two large labeled datasets of e-commerce transactions from two European nations, that have been provided by our industrial partner. The name of the two countries cannot be disclosed: thereby, we will denote

Table 1 Detailed description of the used datasets

Id	Start	End	# Days	# Inst.	Feat.	Fraud %
Country1	02/04/18	30/09/18	181	54809284	25	0.200
Country2	02/04/18	30/09/18	181	30042535	25	0.288

them as Country1 and Country2. Those transactions took place from 02/04/2018 to 30/09/2018, and they sum up to around 50M for Country1 and 30M for Country2. In the considered scenario, the percentage of fraud is exceptionally low, from 0.2% to 0.3%, leading to a *very unbalanced* learning problem.

The original dataset provided by our partner contained 43 features, categorical and numerical, binary and non-binary, discrete and continuous. This dataset is already clean: it does not contain missing values, transactions are labeled, and normalization of the continuous features has been performed. Some features are the results of feature engineering, to extract meaningful aspects of each transactions, but we cannot disclose how.

In order to be coherent with our previous work on the same datasets (Lebichot et al. 2020), we select a subset of 25 meaningful numerical features. Despite the fact that we cannot reveal the true nature of the features, we can say that some describe the accounts, some describe the users, and some describe their behavior. Two of those features are binary, including the target column, while the remaining ones are continuous. A detailed description of the used datasets can be found in Table 1.

In particular it is possible to classify the input features according to the Recency, Frequency, Monetary (RFM) principle (Wei et al. 2010). Relevant examples of application of the RFM principle in the credit card anti-fraud domain can be found in (Baesens 2014; Baesens et al. 2015).

The classification is as follows:

- **Recency**: 4 features are describing the recent events, in terms of average, minimum, maximum, etc.
- **Frequency**: 5 features are counting the number of transactions occurred in the latest *unit* of time, with non-disclosable details.
- **Monetary**: 8 features fall in this category, they describe the intensity of the transaction and other non-disclosable monetary risk measurements.
- **Others**: the remaining input features do not belong in any of the previous categories, and they are mostly describing the client.

It is important to remark that, for the training, all features that uniquely identify a card must be removed, so that the learning model should not take advantage of them. In fact, this would lead to overestimating performance since in real-life a card is blocked after a first fraud is detected.

Synthetic data To address reproducibility, an artificial dataset is also adopted. It is obtained by utilizing a transaction data simulator of legitimate and fraudulent transactions, publicly available at (Le Borgne et al. 2022).

The referenced resource offers a generated dataset of 1754155 transactions and 23 features. Its fraud generation mechanism has been modified to ensure a strong and recurrent concept drift, as follows:

- transactions from the first week of each month are labeled as fraudulent if their amount is between 40 and 50 euros
- transactions from the second week of each month are labeled as fraudulent if their amount is between 50 and 70 euros
- transactions from the third week of each month are labeled as fraudulent if their amount is between 70 and 90 euros
- transactions from the fourth week of each month are labeled as fraudulent if their amount is between 90 and 100 euros

5.3 Dealing with unbalancedness

To address the problem of unbalancedness, we adopt a Balanced Random Forest, which randomly under-samples each boostrap sample to balance it, training each tree with a balanced subsample of the original dataset. It is possible to apply other techniques to address the unbalancedness, and several experiments have been carried out in the past (Dal Pozzolo et al. 2015b), including ADASYN. Easy-ensemble techniques, based on the idea of using different balanced subsamples, have been showing optimal results, and it is very similar to what we have done with the Balanced RF. However, the goal of the paper is not to solve the unbalanced problem, but rather to provide an overlook of the impact of diversity measures and ensemble techniques.

5.4 Model selection

We have mentioned in Sect. 1 that in this work we adopt Random Forest as a model. Specifically, a Balanced Random Forest of 10 trees, where each tree learns from a different balanced subsample of the original dataset. We justify the choice of a Random Forest model with its performance in past fraud detection work (Dal Pozzolo et al. 2014a), but also for the advantage in providing an easy access to feature importance, easing the work of investigators in understanding the reason behind one specific classification. The Random Forest model is also easily adapted to work with unbalanced data (Sect. 5.3). However, this does not imply that simpler models might not be ineffective: recent literature (Baesens et al. 2021) suggests that the performance might depend more on data engineering than complex models. The proposed framework is model-independent, and can be naturally used with other models.

5.5 Experimental settings

In our experiments, we consider a constant verification latency of η days, and according to previous works (Dal Pozzolo et al. 2017), we set $\eta = 7$, being one week considered a valid verification latency size from a practitioner perspective. We also consider a constant size of the training window $\delta = 30$ and in the following we write $\mathcal{K}_{x|30}$

simply as $\mathcal{K}_x$. This choice comes from past work (Dal Pozzolo et al. 2014a) and it usually provides sufficient fraudulent transactions for an effective learning. We choose to set the size of our ensemble $\mathcal{E}$, $s = 7$, chosen as a reasonable value from the results in (Sun et al. 2018), where a similar ensemble is adopted. We test our approaches on batches from B_{67} to B_{126} of our dataset. We consider batches from B_{60} to B_{66} as the gap-set (Fig. 1) due to the verification latency η. Hence, our first trainset for testing on batch B_{67} will cover the batches $\{\mathrm{B}_{59}, \ldots, \mathrm{B}_{30}\}$. We assess the impact of the proposed learning strategy in a gradual way, with increasing complexity. We start from a simple *batch learning*, where a single model is used and no update is performed; we add an update mechanism, based on re-training a single model every single day shifting the trainset, in a *sliding window* manner. Additionally, we introduce the ensemble by storing multiple models and averaging their predictions, using as model selection criteria time, diversity, and randomness. We test all the diversity measures described in Sect. 4.1. Finally, for the ensemble-based approaches we also propose a weighted version, detailed in the following.

Namely, we implement the following approaches, whose variations are summarized in Table 2:

- *Static*: a single model is created once and no update is performed. The classifier evolves as follows:

$$\begin{aligned}\mathcal{K}_{67} &= \mathrm{TRAIN}\left(\{\mathrm{B}_{59}, \mathrm{B}_{58}, \ldots, \mathrm{B}_{30}\}\right)\\ \mathcal{K}_{68} &= \mathcal{K}_{67}\\ &\ldots\end{aligned} \tag{24}$$

- *Sliding Window*: a single model is created every day, using a moving set of batches. The classifier evolves as follows:

$$\begin{aligned}\mathcal{K}_{67} &= \mathrm{TRAIN}\left(\{\mathrm{B}_{59}, \mathrm{B}_{58}, \ldots, \mathrm{B}_{30}\}\right)\\ \mathcal{K}_{68} &= \mathrm{TRAIN}\left(\{\mathrm{B}_{60}, \mathrm{B}_{59}, \ldots, \mathrm{B}_{31}\}\right)\\ &\ldots\end{aligned} \tag{25}$$

- *Recent Ensemble*: the latest 7 *sliding window* models are stored in an ensemble and their average is used for predictions as $\mathcal{P}_{\mathcal{E}}\left(x_i\right) = \frac{1}{7}\sum_{j=1}^{7} \mathcal{P}_{\mathcal{E}[j]}\left(x_i\right)$. The ensemble evolves as follows:

$$\begin{aligned}\mathcal{E}_{67} &= \{\mathcal{K}_{67}, \mathcal{K}_{66}, ..., \mathcal{K}_{61}\}\\ \mathcal{E}_{68} &= \{\mathcal{K}_{68}, \mathcal{K}_{67}, ..., \mathcal{K}_{62}\}\\ &\ldots\end{aligned} \tag{26}$$

  An additional weighted version is proposed to favor recent models, the only difference is $\mathcal{P}_{\mathcal{E}}\left(x_i\right) = \sum_{j=1}^{7} w_j \mathcal{P}_{\mathcal{E}[j]}\left(x_i\right)$ where w_j is simply j.

- *Random Ensemble*: after an initialization phase where the latest 7 *sliding window* models are stored in an ensemble, new models are added to the ensemble and one member is randomly removed at each iteration. Predictions are computed as $\mathcal{P}_{\mathcal{E}}\left(x_i\right) = \frac{1}{7}\sum_{j=1}^{7} \mathcal{P}_{\mathcal{E}[j]}\left(x_i\right)$. The ensemble evolves as follows:

Table 2 A summary of all tested approaches

Full name	Identifier	Type	Update criterion	Weights
Static	*static*	Single	None	None
Sliding Window	*slidingwin*	Single	Time	None
Recent Ensemble	*rec_ens*	Ensemble	Time	None
Recent Weighted Ensemble	*w_rec_ens*	Ensemble	Time	Time
Random Ensemble	*random_ens*	Ensemble	Random	None
Diverse Q Ensemble	*divQ_ens*	Ensemble	Yule's Q	None
Diverse RO Ensemble	*divRO_ens*	Ensemble	Correlation coeff.	None
Diverse DIS Ensemble	*divDIS_ens*	Ensemble	Disagreement	None
Diverse DF Ensemble	*divDF_ens*	Ensemble	Double Fault	None
Diverse IA Ensemble	*divIA_ens*	Ensemble	Interr. Agreement	None
Diverse Weighted Q Ensemble	*w_divQ_ens*	Ensemble	Yule's Q	Pk100
Diverse Weighted RO Ensemble	*w_divRO_ens*	Ensemble	Correlation coeff.	Pk100
Diverse Weighted DIS Ensemble	*w_divDIS_ens*	Ensemble	Disagreement	Pk100
Diverse Weighted DF Ensemble	*w_divDF_ens*	Ensemble	Double Fault	Pk100
Diverse Weighted IA Ensemble	*w_divIA_ens*	Ensemble	Interr. Agreement	Pk100

$$\begin{aligned} \mathcal{E}_{67} &= \{\mathcal{K}_{67}, \mathcal{K}_{66}, ..., \mathcal{K}_{61}\} \\ \mathcal{E}_{68} &= (\mathcal{E}_{67} \cup \{\mathcal{K}_{68}\}) \setminus \mathcal{E}[j], \text{j is random} \\ &\quad \dots \end{aligned} \tag{27}$$

- *Diverse Ensemble*: after an initialization phase where the latest 7 *sliding window* models are stored in an ensemble, new models are added to the ensemble and one member is removed by maximizing a diversity measure computed on unseen data. Predictions are computed as $\mathcal{P}_{\mathcal{E}}(x_i) = \frac{1}{7}\sum_{j=1}^{7} \mathcal{P}_{\mathcal{E}[j]}(x_i)$. The ensemble evolves as follows:

$$\begin{aligned} \mathcal{E}_{67} &= \{\mathcal{K}_{67}, \mathcal{K}_{66}, ..., \mathcal{K}_{61}\} \\ \mathcal{E}_{68} &= (\mathcal{E}_{67} \cup \{\mathcal{K}_{68}\}) \setminus \mathcal{E}[j], \text{j maximises diversity} \\ &\quad \dots \end{aligned} \tag{28}$$

  An additional weighted version is proposed to favor performing models, the only difference is $\mathcal{P}_{\mathcal{E}}(x_i) = \sum_{j=1}^{7} w_j \mathcal{P}_{\mathcal{E}[j]}(x_i)$ where w_j is computed as in (13), using $\mathcal{M}$ =`pk100`.

5.6 Results

We present here the metrics mentioned in Sect. 5.1: Precision Top 100 (`pk100`) and Calibrated Area Under the Precision Recall Curve with *card grouping* (`cprauc_c`). We present the collected metrics for both countries in Table 3: we show *mean* and

Table 3 Fraud detection performance when using 30 days of transactions as the train-set, with an ensemble of size 7 and a *verification latency* of 7 days ($s = 7, \eta = 7, \delta = 30$)

		Country1		Country2	
		pk100	*cprauc_c*	*pk100*	*cprauc_c*
Static	Mean	0.106	0.159	0.252	0.101
	Std	0.047	0.075	0.207	0.049
Slidingwin	Mean	0.144	0.177	0.244	0.092
	Std	0.061	0.049	0.169	0.024
rec_ens	Mean	0.307	**0.280**	0.513	**0.180**
	Std	0.091	0.082	0.287	0.059
w_rec_ens	Mean	**0.309**	**0.278**	0.513	0.178
	Std	0.093	0.080	0.284	0.059
divQ_ens	Mean	**0.309**	0.265	0.516	0.178
	Std	0.079	0.093	0.273	0.058
divRO_ens	Mean	0.304	0.265	0.511	0.177
	Std	0.084	0.093	0.270	0.056
divDIS_ens	Mean	0.304	0.266	0.505	0.176
	Std	0.081	0.095	0.266	0.055
divDF_ens	Mean	**0.311**	**0.281**	**0.525**	**0.181**
	Std	0.086	0.083	0.285	0.057
divIA_ens	Mean	0.307	0.260	0.509	0.179
	Std	0.090	0.098	0.273	0.057
w_divQ_ens	Mean	0.307	0.267	**0.517**	0.178
	Std	0.078	0.094	0.274	0.058
w_divRO_ens	Mean	0.306	0.267	0.503	0.177
	Std	0.083	0.096	0.266	0.055
w_divDIS_ens	Mean	0.303	0.266	**0.520**	**0.179**
	Std	0.080	0.094	0.279	0.057
w_divDF_ens	Mean	**0.308**	**0.281**	**0.526**	**0.181**
	Std	0.083	0.084	0.287	0.058
w_divIA_ens	Mean	0.305	0.261	**0.527**	**0.182**
	Std	0.082	0.098	0.284	0.059
random_ens	Mean	**0.311**	**0.275**	**0.517**	0.178
	Std	0.089	0.086	0.280	0.058

The top-5 scoring approaches for each metric are in bold. It is possible to see that some approaches (e.g. *divDF_ens*) belong to the top-5 for all metrics presented

standard deviation computed over the test days. The top-5 approaches for each metric are in bold. To assess the statistical significance of our results we perform Friedman/Nemenyi tests ($\alpha = 0.05$) as recommended by (Demšar 2006) and we reject the null hypothesis that all classifiers achieve the same performance. The results of the statistical tests are presented by means of a Critical Diversity plot. Fig. 2 shows the Critical Diversity plot to compare the various diversity measures, while Fig. 3 presents

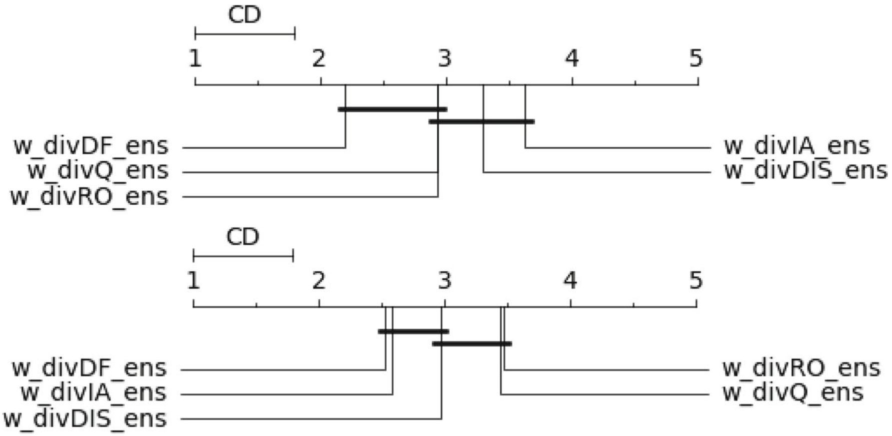

Fig. 2 Friedman/Nemenyi test for models comparison (the lower, the better)—diversity measures comparison for cprauc_card metric. Country1 above, Country2 below. Approaches connected with a bold line are statistically equivalent

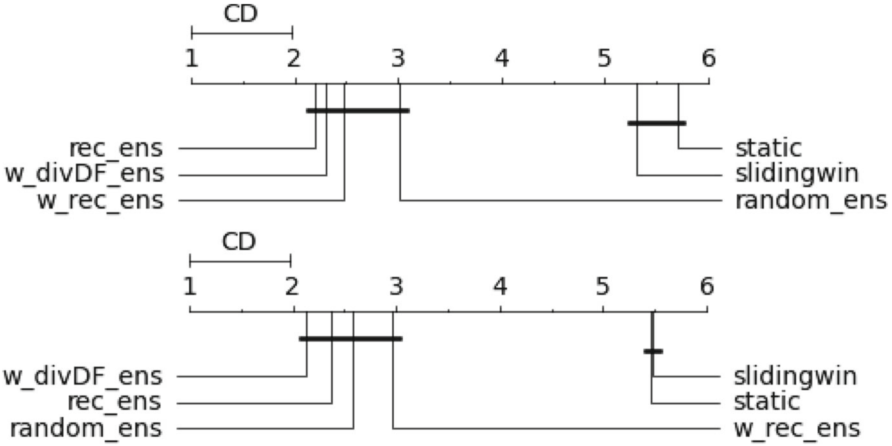

Fig. 3 Friedman/Nemenyi test for models comparison (the lower, the better)—approaches comparison for cprauc_card metric. Country1 above, Country2 below. Approaches connected with a bold line are statistically equivalent

the Critical Diversity plot comparing the best diversity measure from Fig. 2 with other approaches. To visually compare the most performing approaches, we plot the evolution of one of the metrics (`pk100`) over time in Fig. 4. The evolution of the metric `pk100` over time obtained on the artificial dataset described in Sect. 5.2 can be found in Fig. 5

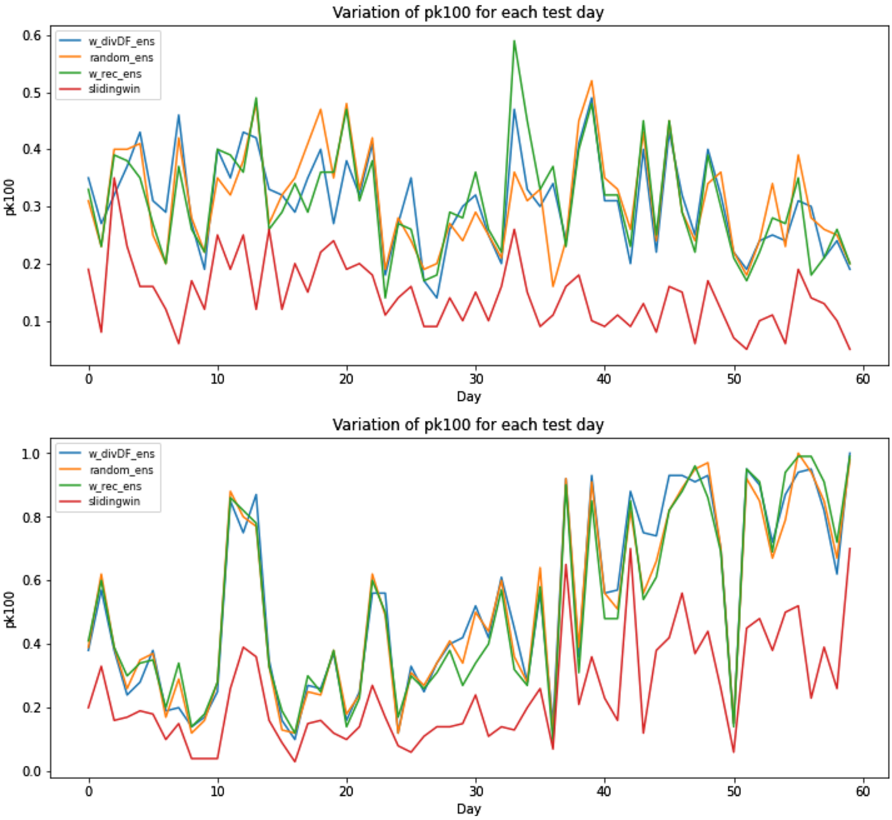

Fig. 4 Evolution of the pk100 metric for the top scoring approaches over the test days (the higher, the better). Country1 above, Country2 below

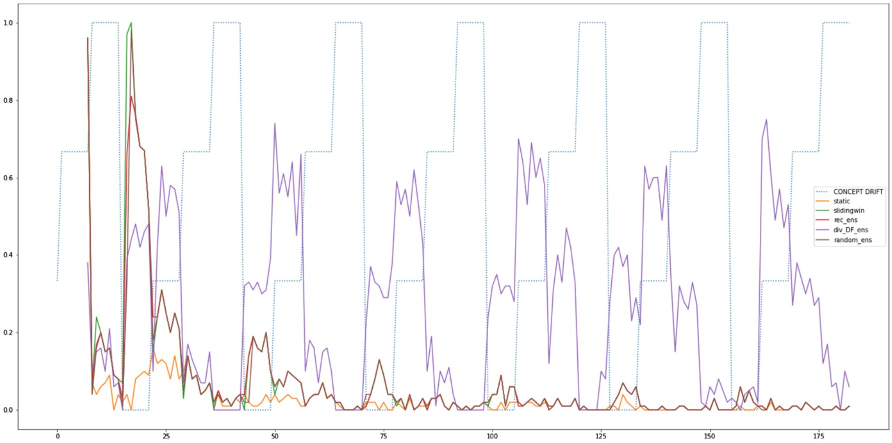

Fig. 5 Evolution of the pk100 metric for a selection of approaches over the test days of the artificial dataset described in Sect. 5.2

5.7 Discussion

On the basis of the above experiments these are the main considerations:

- Figure 2 shows that using different diversity measures can affect the performance of the ensemble. In particular, we see that there is a significant difference between the best diversity measures and the worst ones. This suggests special care when choosing the diversity measure. Double Fault measure appears to be the best choice for both countries, while the other diversity measures obtain different rankings. Double Fault (18) (Giacinto and Roli 2001) computes the fraction of times when a pair of classifier is wrong, i.e. both classifiers wrongly predict a class. We can speculate that, being the fraud detection problem a complex task (see Sect. 1), a measure proportional to the *mistakes* of both classifiers can help to differentiate them. The different rankings obtained by the other diversity measure suggest that an ideal diversity measure might depend on the data distribution.
- Comparing the Double Fault diversity-based approach with the other approaches summarized in Table 1, it is quickly noticeable that ensemble-based models overperform single models (Fig. 3) even when the approach is quite similar, as it is for `rec_ens` and `slidingwin` (the former is an ensemble variant of the latter). In this case, weighting to favor the *most recent* model in the ensemble appears detrimental rather than beneficial: this-together with the `slidingwin` bad performance—confirms the value of historical knowledge and discourage an exclusive focus on recent data. However, `rec_ens` has shown to win—although not significantly (Fig. 3)—over the diversity-based approach for Country1. This is a valid example of the *stability-plasticity dilemma*: exclusive usage of recent data is not optimal, but recent data does play a crucial role in the performance of the model.
- From the statistical tests in Fig. 3, the ensemble approaches do not show any significant difference among them. This can be grasped also by looking at the evolution of the metrics over time (Fig. 4): lines are almost overlapping for most days, but they present interesting peaks. For example in Country2 (Fig. 3, below), around Day 30 and from Day 40 to 50, the `w_divDF_ens` line is above the `rec_ens` line with a non-negligible difference (~ 0.2), showing a possible improved adaptation by exploiting historical knowledge.
- Additionally, in Fig. 3 it is interesting to notice the behavior of the `random_ens` approach: it performs much better than expected. This can be motivated by the fact that adding randomness can be considered as a way of increasing diversity: if we update randomly a uniform ensemble, we will be likely decreasing its uniformity. The final ensemble benefits from randomness as it will potentially contain unrelated models after several iterations.
- The results on the artificial dataset in Fig. 5 confirm that in presence of a recurrent concept drift the diversity-based ensemble approach is significantly overperforming all the other approaches.

Further experiments, not reported here, have shown no significant difference between weighted versions of diversity-based ensemble and non-weighted ones, and often the same diversity measure has prevailed in both its weighted and non-weighted variants over the other diversity measures. This suggests that *which* diversity measure is used to select the stored classifiers might be more important than the way the classifiers are combined.

6 Conclusion

In this work, we propose the use of a diversity-based ensemble approach to improve the performance of a credit card transactions classifier for fraud detection. The intuition behind this approach is that the evolving nature of transactions would favor the presence of recurring concepts, whose associated knowledge is strictly connected to the past. Having a *memory of past knowledge* as an ensemble of models could ease concept drift adaptation especially when a past concept re-appears. We compared several individual and ensemble approaches over an extensive real-life dataset, by adopting diversity-based, time-based and random measures to choose between the ensemble candidates. Our goal was to understand (1) if diversity is a valid criterion to build the ensemble, (2) what is the impact of different diversity measures and, (3) if our proposed learning strategy is effective. The results have shown that (1) diversity is a valid criterion, but it is not significantly better than other criteria like *recentness*, (2) varying diversity measure does impact the results and the best diversity measure, for us the Double Fault measure (18), is probably problem-dependent and (3) our proposed learning strategy ends up in the set of the top scoring approaches, although not uniquely, confirming its effectiveness. Further work will study the impact of varying parameters, and will assess the role of Diversity by taking into account the *alert-feedback interaction* (Dal Pozzolo et al. 2017). Additionally, it will extend the set of diversity measures considered, including non-pairwise measures.

Funding This work was supported by the TeamUp DefeatFraud project funded by Innoviris (2017-R-49a). We thank this agency for allowing us to conduct both fundamental and applied research. Gian Marco Paldino and Gianluca Bontempi are supported by the Service Public de Wallonie Recherche under grant nr 2010235–ARIAC by DigitalWallonia4.ai. Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

Availability of data, material and code: The real data and original code cannot be made available for confidentiality reasons. To ensure reproducibility, a preliminary version of the code applied to the synthetic data used in the manuscript, is available at the following link: https://github.com/gmpal/diversity_ensemble_fraud_detection. The code will be extended and improved in the future to fully cover the manuscript experiments. The repository also contains the synthetic data. The original synthetic dataset and its generator can be found in (Le Borgne et al., 2022): an open book published by the ULB MLG group about reproducible machine learning in fraud detection. Additionally, the book provides open implementations of fraud detection functions and performance metrics like the ones presented in this manuscript. Furthermore, similar dataset is available on Kaggle (Machine Learning Group-ULB, 2018): it is a two-day long, anonymized extract from the same database.

Declarations

Conflict of interest The authors declare they have no conflicts of interest or competing interests.

References

Accenture (2020) How covid-19 will permanently change consumer behavior

Alazizi A, Habrard A, Jacquenet F, He-Guelton L, Oblé F, Siblini W (2019) Anomaly detection, consider your dataset first an illustration on fraud detection. In: 2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI). IEEE, pp 1351–1355

Alippi C, Boracchi G, Roveri M (2013) Just-in-time classifiers for recurrent concepts. IEEE Trans Neural Netw Learn Syst 24(4):620–634

Ba H (2019) Improving detection of credit card fraudulent transactions using generative adversarial networks. arXiv:1907.03355

Baesens B (2014) Analytics in a big data world: the essential guide to data science and its applications. Wiley, New York

Baesens B, Van Vlasselaer V, Verbeke W (2015) Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection. Wiley, New York

Baesens B, Höppner S, Verdonck T (2021) Data engineering for fraud detection. Decis Support Syst 150:113492

Batista GE, Carvalho AC, Monard MC (2000) Applying one-sided selection to unbalanced datasets. In: Mexican international conference on artificial intelligence. Springer, pp 315–325

Benesty J, Chen J, Huang Y, Cohen I (2009) Pearson correlation coefficient. In: Noise reduction in speech processing. Springer, pp 1–4

Bhattacharyya S, Jha S, Tharakunnel K, Westland JC (2011) Data mining for credit card fraud: a comparative study. Decis Support Syst 50(3):602–613

Bolton RJ, Hand DJ (2002) Statistical fraud detection: a review. Stat Sci 17(3):235–255

Bolton RJ, Hand DJ et al (2001) Unsupervised profiling methods for fraud detection. In: Credit scoring and credit control VII, pp 235–255

Brause R, Langsdorf T, Hepp M (1999) Neural data mining for credit card fraud detection. In: Proceedings 11th international conference on tools with artificial intelligence. IEEE, pp 103–106

Breiman L, Friedman JH, Olshen RA, Stone CJ (2017) Classification and regression trees. Routledge, London

Carcillo F, Le Borgne YA, Caelen O, Kessaci Y, Oblé F, Bontempi G (2019) Combining unsupervised and supervised learning in credit card fraud detection. Inf Sci 557:317–331

Cerasa A, Cerioli A (2017) Outlier-free merging of homogeneous groups of pre-classified observations under contamination. J Stat Comput Simul 87(15):2997–3020. https://doi.org/10.1080/00949655.2017.1351564

Cerioli A, Barabesi L, Cerasa A, Menegatti M, Perrotta D (2018) Newcomb–benford law and the detection of frauds in international trade. Proc Natl Acad Sci 116(1):106–115. https://doi.org/10.1073/pnas.1806617115

Chan PK, Fan W, Prodromidis AL, Stolfo SJ (1999) Distributed data mining in credit card fraud detection. IEEE Intell Syst Their Appl 14(6):67–74

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Clark P, Niblett T (1989) The cn2 induction algorithm. Mach Learn 3(4):261–283

Cunningham P, Carney J (2000) Diversity versus quality in classification ensembles based on feature selection. In: European conference on machine learning. Springer, pp 109–116

Dal Pozzolo A (2015) Adaptive machine learning for credit card fraud detection

Dal Pozzolo A, Caelen O, Waterschoot S, Bontempi G (2013) Racing for unbalanced methods selection. In: International conference on intelligent data engineering and automated learning. Springer, pp 24–31

Dal Pozzolo A, Caelen O, Le Borgne YA, Waterschoot S, Bontempi G (2014a) Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl 41(10):4915–4928

Dal Pozzolo A, Johnson R, Caelen O, Waterschoot S, Chawla NV, Bontempi G (2014b) Using hddt to avoid instances propagation in unbalanced and evolving data streams. In: 2014 International joint conference on neural networks (IJCNN). IEEE, pp 588–594

Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G (2015a) Credit card fraud detection and concept-drift adaptation with delayed supervised information. In: 2015 international joint conference on Neural networks. IEEE

Dal Pozzolo A, Caelen O, Bontempi G (2015b) When is undersampling effective in unbalanced classification tasks? In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 200–215

Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G (2017) Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Trans Neural Netw Learn Syst 29(8):3784–3797

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(Jan):1–30

Domingo C, Gavalda R, Watanabe O (2002) Adaptive sampling methods for scaling up knowledge discovery algorithms. Data Min Knowl Disc 6(2):131–152

Dorronsoro JR, Ginel F, Sgnchez C, Cruz CS (1997) Neural fraud detection in credit card operations. IEEE Trans Neural Netw 8(4):827–834

Drummond C, Holte RC, et al (2003) C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. Workshop on learning from imbalanced datasets II, Citeseer 11, pp 1–8

Fleiss JL, Levin B, Paik MC et al (1981) The measurement of interrater agreement. Stat Methods Rates Proportions 2(212–236):22–23

Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In: Brazilian symposium on artificial intelligence. Springer, pp 286–295

Gama J, Žliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014a) A survey on concept drift adaptation. ACM Comput Surv 46(4):1–37. https://doi.org/10.1145/2523813

Gama J, Žliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014b) A survey on concept drift adaptation. ACM Comput Surv (CSUR) 46(4):1–37

Giacinto G, Roli F (2001) Dynamic classifier selection based on multiple classifier behaviour. Pattern Recogn 34(9):1879–1882

Haibo H, Yang B, Edwardo GA, Shutao L (2016) Adaptive synthetic sampling approach for imbalanced learning. IEEE Int Joint Conf Neural Netw IJCNN 8:1322–1328

Hand DJ (1981) Discrimination and classification. Wiley Series in Probability and Mathematical Statistics

He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284

Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844

Hodge V, Austin J (2004) A survey of outlier detection methodologies. Artif Intell Rev 22(2):85–126

Holte RC, Acker L, Porter BW et al (1989) Concept learning and the problem of small disjuncts. IJCAI Citeseer 89:813–818

Jha S, Guillen M, Westland JC (2012) Employing transaction aggregation strategy to detect credit card fraud. Expert Syst Appl 39(16):12650–12657

Juszczak P, Adams NM, Hand DJ, Whitrow C, Weston DJ (2008) Off-the-peg and bespoke classifiers for fraud detection. Comput Stat Data Anal 52(9):4521–4532

Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A et al (2017) Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci 114(13):3521–3526

Krogh A, Vedelsby J (1995) Validation, and active learning. Adv Neural Inf Process Syst 7(7):231

Kuncheva LI (2004) Classifier ensembles for changing environments. In: International workshop on multiple classifier systems. Springer, pp 1–15

Le Borgne YA, Siblini W, Lebichot B, Bontempi G (2022) Reproducible machine learning for credit card fraud detection-practical handbook. Université Libre de Bruxelles. https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook

Lebichot B, Braun F, Caelen O, Saerens M (2016) A graph-based, semi-supervised, credit card fraud detection system. In: International workshop on complex networks and their applications. Springer, pp 721–733

Lebichot B, Le Borgne YA, He-Guelton L, Oblé F, Bontempi G (2019) Deep-learning domain adaptation techniques for credit cards fraud detection. In: INNS big data and deep learning conference. Springer, pp 78–88

Lebichot B, Paldino G, Bontempi G, Siblini W, He-Guelton L, Oblé F (2020) Incremental learning strategies for credit cards fraud detection: extended abstract. In: 2020 IEEE 7th international conference on data science and advanced analytics (DSAA), pp 785–786. https://doi.org/10.1109/DSAA49011.2020.00116

Liu XY, Wu J, Zhou ZH (2008) Exploratory undersampling for class-imbalance learning. IEEE Trans Syst Man Cybern Part B (Cybern) 39(2):539–550

Mermillod M, Bugaiska A, Bonin P (2013) The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. Front Psychol 4:504

Mullick SS, Datta S, Das S (2019) Generative adversarial minority oversampling. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1695–1704

Phua C, Lee V, Smith K, Gayler R (2010) A comprehensive survey of data mining-based fraud detection research. arXiv:1009.6119

Pourhabibi T, Ong KL, Kam BH, Boo YL (2020) Fraud detection: a systematic literature review of graph-based anomaly detection approaches. Decis Support Syst 133:113303

Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND (2009) Dataset shift in machine learning. The MIT Press, New York

Rousseeuw P, Perrotta D, Riani M, Hubert M (2019) Robust monitoring of time series with application to fraud detection. Econom Stat 9:108–121. https://doi.org/10.1016/j.ecosta.2018.05.001

Schlimmer JC, Granger RH (1986) Incremental learning from noisy data. Mach Learn 1(3):317–354

Siblini W, Fréry J, He-Guelton L, Oblé F, Wang YQ (2020) Master your metrics with calibration. In: International symposium on intelligent data analysis. Springer, pp 457–469

Sun Y, Tang K, Zhu Z, Yao X (2018) Concept drift adaptation by exploiting historical knowledge. IEEE Trans Neural Netw Learn Syst 29(10):4822–4832

Van Vlasselaer V, Bravo C, Caelen O, Eliassi-Rad T, Akoglu L, Snoeck M, Baesens B (2015) Apate: a novel approach for automated credit card transaction fraud detection using network-based extensions. Decis Support Syst 75:38–48

Veeramachaneni K, Arnaldo I, Korrapati V, Bassias C, Li K (2016) Ai2: training a big data machine to defend. In: 2016 IEEE 2nd international conference on big data security on cloud (BigDataSecurity), pp 49–54

Washio T, Motoda H (2003) State of the art of graph-based data mining. ACM SIGKDD Explor Newsl 5(1):59–68

Webb AR (2003) Statistical pattern recognition. Wiley, New York

Wei JT, Lin SY, Wu HH (2010) A review of the application of rfm model. Afr J Bus Manag 4(19):4199–4206

Weston DJ, Hand DJ, Adams NM, Whitrow C, Juszczak P (2008) Plastic card fraud detection using peer group analysis. Adv Data Anal Classif 2(1):45–62

Whitrow C, Hand DJ, Juszczak P, Weston D, Adams NM (2009) Transaction aggregation as a strategy for credit card fraud detection. Data Min Knowl Disc 18(1):30–55

Widmer G, Kubat M (1996) Learning in the presence of concept drift and hidden contexts. Mach Learn 23(1):69–101

Yule GU (1900) Vii. on the association of attributes in statistics. Philos Trans Roy Soc Lond 194(252–261):257–319

Zhou ZH (2009) Ensemble learning. Encycl Biom 1:270–273

Žliobaite I (2010) Learning under concept drift: an overview. arXiv:1010.4784

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.