# Journal Pre-proof


Assessment of catastrophic forgetting in continual credit card fraud detection

B. Lebichot, W. Siblini, G.M. Paldino, Y.-A. Le Borgne, F. Oblé, G. Bontempi

PII: S0957-4174(24)00310-5
DOI: https://doi.org/10.1016/j.eswa.2024.123445
Reference: ESWA 123445

To appear in: *Expert Systems With Applications*

Received date : 11 June 2021
Revised date : 29 September 2022
Accepted date : 7 February 2024

Please cite this article as: B. Lebichot, W. Siblini, G.M. Paldino et al., Assessment of catastrophic forgetting in continual credit card fraud detection. *Expert Systems With Applications* (2024), doi: https://doi.org/10.1016/j.eswa.2024.123445.

Highlights:

- Fraud detection models must be updated continually to handle new fraud strategies.
- They must balance plasticity (learn new patterns) and stability (remember old ones).
- We show how to quantify both and discuss the trade-off for fraud detection.
- We provide an extensive comparison of six strategies and 13 different models.
- We present a real case study based on more than 50 million e-commerce transactions.

# Assessment of catastrophic forgetting in continual credit card fraud detection

B. Lebichot[a,*], W. Siblini[b], G. M. Paldino[a], Y.-A. Le Borgne[a], F. Oblé[b], G. Bontempi[a]

[a] *Machine Learning Group, Computer Science Departement, Faculty of Sciences, Université Libre de Bruxelles, Belgium.*
[b] *Research, Development and Innovation, Worldline, Lyon.*

*Key words:* Catastrophic forgetting, Fraud detection, Incremental learning, Continual learning, Continuous learning, Fintech

*Corresponding author.
*Email addresses:* bertrand.lebichot@gmail.com (B. Lebichot ), wissam.siblini92@gmail.com (W. Siblini), gian.marco.paldino@ulb.be (G. M. Paldino), yann-ael.le.borgne@ulb.be (Y.-A. Le Borgne), frederic.oble@worldline.com (F. Oblé), gianluca.bontempi@ulb.be (G. Bontempi)

**Abstract**

The volume of e-commerce continues to increase year after year. Buying goods on the internet is easy and practical, and took a huge boost during the lockdowns of the Covid crisis. However, this is also an open window for fraudsters and the corresponding financial loss costs billions of dollars. In this paper, we study e-commerce credit card fraud detection, in collaboration with our industrial partner, Worldline. Transactional companies are more and more dependent on machine learning models such as deep learning anomaly detection models, as part of real-world fraud detection systems (FDS). We focus on continual learning to find the best model with respect to two objectives: to maximize the accuracy and to minimize the catastrophic forgetting phenomenon. For the latter, we proposed an evaluation procedure to quantify the forgetting in data streams with delayed feedback: the plasticity/stability visualization matrix. We also investigated six strategies and 13 methods on a real-size case study including five months of e-commerce credit card transactions. Finally, we discuss how the trade-off between plasticity and stability is set, in practice, in the case of FDS.

## 1. Introduction

Fraud Detection Systems (FDS) are expert systems that are confronted with streams of credit card transactions and aim to discriminate between fraudulent and genuine transactions. Conventional FDS rely on slowly evolving expert rules Kou et al. (2004); Whitrow et al. (2009) which require important human resources and risk missing complex fraud patterns. Recent FDS evolved towards hybrid configurations combining expert knowledge and machine learning algorithms, notably batch algorithms based on decision trees or neural networks.

These systems have four well-known challenges: (i) the magnitude of the number of transactions to process is in millions per day, (ii) the genuine and the fraudulent class are strongly unbalanced (fraud detection can also be seen as an instance of supervised anomaly detection Chalapathy & Chawla (2019); Pang et al. (2020)), (iii) feedback from human investigators Dal Pozzolo et al. (2015) must be integrated, but have a delay of several days or weeks, and (iv) such systems require adaptive strategies to deal with fraud nonstationary issues.

This paper focus on this last constraint, without discarding the three others. As examples of nonstationnarities, think about the following: the set of cardholders and terminals evolves over time Alazizi et al. (2019), the customer behavior is prone to drift (change of habits, seasonality, and events occurrence) and, last but not least, fraudsters adapt their criminal strategies. It is therefore crucial for FDS to include some strategies to be adaptive (to integrate new fraud patterns), but also avoid forgetting the already observed patterns of fraud.

Learning systems in nature are inherently incremental and robust to forgetting Cichon & Gan (2015); Losing et al. (2018). They acquire knowledge over time by experiencing novel situations and use biological mechanisms to retain it for later use. A similar functionality is also desirable in artificial learning systems, and a lot of attention has been put into research over the past few years Kirkpatrick et al. (2017); Li & Hoiem (2017); Parisi et al. (2019); Shin et al. (2017); Zenke et al. (2017).

The simplest strategy to make batch algorithms adaptive consists of periodically retraining them with a fixed-size sliding window of transactions Dal Pozzolo et al. (2017); Ditzler & Polikar (2012). This strategy is expensive in time and storage (a single month of data requires tens of RAM Gigabytes) and has a legal drawback since data protection regulations (e.g. the European General Data Protection Regulation (GDPR)) impose time limits for the storage of transactional data. An alternative consists in adopting incremental learning machines (e.g. neural networks) and updating them as soon as new transactions are made available Sahoo et al. (2018). Unfortunately, neural networks (NN) are prone to the *catastrophic forgetting* Kirkpatrick et al. (2017); Li & Hoiem

(2017), denoting the fact that the model tends to forget concepts, which did not appear in the training examples for a long time. This problem is of prime importance in credit card fraud detection since decisive events (e.g. holidays or leakage on social media) occur rarely or randomly throughout the year.

To mitigate this catastrophic forgetting issue, research works in *continual and lifelong learning* Parisi et al. (2019); Shin et al. (2017); Zenke et al. (2017) investigates the trade-off between *plasticity* (learning new tasks) and *stability* (remembering old knowledge) Mermillod et al. (2013) by regularizing the model or replaying old memories. However only a few papers deal with other tasks than image recognition, and none of them address the fraud detection case.

One of the goals of this paper is to bridge this gap and quantify the catastrophic forgetting in the case of a massive, non-stationary, real data fraud detection scenario. Indeed, while the assessment of plasticity is quite common in the literature about concept drift, no well-established procedure for assessing the stability of fraud detection learning techniques is available. In literature, most papers rely on artificial splits of well-known datasets (e.g. MNIST) Goodfellow et al. (2013),Zenke et al. (2017),Kirkpatrick et al. (2017) or *data permutation experiments* Kemker et al. (2018). In fact, none of those assessment strategies is convenient for the time-varying/streaming nature of fraud detection.

In this paper, in order to address the lack of results and methodology for addressing continual learning in a realistic fraud detection setting:

1. we design and run several of incremental learning strategies (based on neural networks) over a massive, unbalanced, drifting, dataset of credit-card transactions provided by the industrial partner,
2. we summarize and assess the performance of those algorithms through a matrix visualization, designed for fraud detection, able to decompose the impact of plasticity and stability on the final precision,
3. we explore one solution to the plasticity/stability trade-off in the case of fraud detection based on a set of requirements provided by domain experts (other trade-offs are of course possible).

The rest of this paper is structured as follows: Section 2 discusses some background and related work on continual learning. Section 3 introduces important peculiarities of real-world FDS. Section 4 describes the investigated methods. Section 5 details the plasticity/stability visualization matrix. Section 6 presents the experimental assessments we conducted. Finally, Section 7 concludes the paper.

## 2. Continual learning background

Continual Learning (CL) Goodfellow et al. (2013); Parisi et al. (2019) aims to learn from a stream of data by ensuring: (i) *adaptability*, i.e. models are updated (e.g. with backpropagation) as soon as new data are available and (ii) *scalability*, i.e. the ability to cope with high-dimensional data streams. Unfortunately, adaptability may occur at the cost of *catastrophic forgetting* French (1999); McCloskey & Cohen (1989); Ratcliff (1990), which can be seen as a lack of stability. Multiple solutions to catastrophic forgetting have been proposed Kemker et al. (2018); Maltoni & Lomonaco (2019); Zenke et al. (2017). One of the main contributions of this paper is that it screens all families of approaches, in the particular case of fraud detection, and investigates the five following directions:

- *Rehearsal strategies* periodically replay past observations to reinforce model connections encoding old concepts McCloskey & Cohen (1989); Ratcliff (1990). A simple approach is to store a portion of the past training data and interleaving them with fresh ones during training. Several variants have been proposed: recency rehearsal (based on time), random rehearsal, and sweep rehearsal (dynamic rather than fixed buffer). An important distinction exists between rehearsal strategies and pseudo-rehearsal strategies Robins (1995). In the latter, previous training data are unavailable (e.g. because of data protection regulations) and replaced by synthetically generated samples Hayes et al. (2019); Riemer et al. (2018); Shin et al. (2017) (e.g. by GAN). More recently, the authors of van de Ven

et al. (2020) propose a variant of generative replay inspired by the biological reactivation of neuronal activity patterns, in which internal or hidden representations are replayed that are generated by the network's own, context-modulated feedback connections. Our Generative Replay (*GenRE*) approach is closely related to this approach.

- *Regularization strategies* modify the loss function to promote selective consolidation of the weights which are important to retain past memories Kirkpatrick et al. (2017). They use standard regularization techniques such as weight sparsification, dropout, or early stopping. Well-known regularization-based strategies are elastic weight consolidation (EWC) and incremental moment matching (IMM) Kemker et al. (2018). Other examples are Learning without Forgetting (LwF) Li & Hoiem (2017)), knowledge distillation Parisi et al. (2019) and synaptic intelligence (SI) Zenke et al. (2017). Recently, authors of Li et al. (2021) propose a Sketched Structural Regularization leveraging linear sketching methods as an alternative approach to compress the importance matrices used for regularizing in Structural Regularization methods.

- *Architectural strategies* use specific architectures, layers, activation functions, model expansions, and/or weight-freezing strategies (e.g. CWR Lomonaco & Maltoni (2017)) to mitigate catastrophic forgetting Polikar et al. (2001); Sun et al. (2018). Progressive Neural Networks (PNN) combine parameter freezing and network expansion Rusu et al. (2016).

Another important distinction can be made between multi-task and single-task problems Maltoni & Lomonaco (2019).

- MT: Multi-Task problems have to deal with a succession of isolated tasks (for example, if we want to learn an additional task on top of a pre-trained network) and avoid forgetting the previous ones. Though this approach is intuitive, it is difficult to adopt in real-life incremental settings where the decomposition in separate concepts/tasks is not always straightforward.

- SIT Sun et al. (2018): Single-Incremental-Tasks aim to deal with a unique task evolving with three possible kinds of updates:
  - New Instances (NI): new training patterns of the same classes become available in subsequent chunks with new environment conditions. For instance, in FDS this occurs when new behaviors appear (e.g. increased e-commerce activity during the lockdown or a new fraudster strategy)
  - New Classes (NC): new training patterns belonging to different classes become available in subsequent chunks. This is an example of class-incremental learning Rebuffi et al. (2017).
  - New Instances and Classes (NIC): new training patterns belonging to both known and new classes become available in subsequent training chunks.

A more recent categorization of the contribution in the literature and review the existing benchmarks is proposed in Cossu et al. (2021). Authors also provide two new benchmarks for CL with sequential data based on existing datasets, as well as a broad empirical evaluation of CL and Recurrent Neural Networks in class-incremental scenario. They show that key factors are the sequence length and a clear specification of the CL scenario. Additionally, authors of Ramasesh et al. (2021) show that pretrained models such as ResNets and Transformers are significantly less prone to forgetting with respect to models that are randomly initialized and trained from scratch. The resistance to forgetting is proportional to the scale of the model and the pretraining dataset size, which are, together with a diverse pretraining dataset, useful ingredients in mitigating the problem.

This paper focuses on a detection task which belongs to the SIT-NI case since the number of classes is fixed (fraudulent vs. genuine) and most variability is due to concept drift.

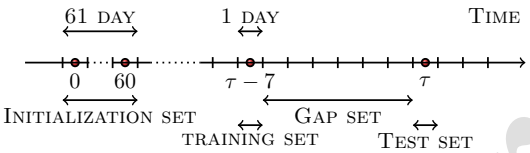


Figure 1: Splitting strategy for the initialization/training/gap/test sets. In this paper, the labeled training set is composed of chunk of one day and the unlabeled transactions of the gap set are simply discarded. After each update of the model, the training, gap, and test sets slide one day to the right ($\tau$ is increased by one).

## 3. Real-world FDS

This paper focuses on real-world FDS dealing with streams of credit card transactions. A FDS raises alerts on the basis of a risk score computed as the conditional probability of a fraudulent transaction. The raised alerts are subsequently checked by human investigators who decide about the most relevant action (e.g. the card is blocked, the card holder is contacted,...) Dal Pozzolo et al. (2014). Given that human investigators are an expensive and scarce resource, the FDS assessment is done with respect to the accuracy of a limited number (typically $k = 100$) of high risk alerts. Measures like Pr@$k$, the *precision* over the first $k$ alerts, are commonly used Dal Pozzolo et al. (2014). Accuracy measures of the entire ranking like the *area under the precision-recall curve* (AUPRC) have been considered in litterature as well Davis & Goadrich (2006); Saito & Rehmsmeier (2015); Siblini et al. (2020).

Real-world FDS Carcillo et al. (2018) typically group transactions into *chunks* according to a size or temporal criterion. In our experiments we consider chunks at daily level for the following reasons: (i) it is the production setting adopted by the industrial partner, since performance metrics make more sense at daily level, (ii) a preliminary study Lebichot et al. (2020) shows daily updates are competitive in terms of performance, and (iii) higher frequency could be detrimental for the model since the sub-daily (e.g. hourly) distribution of frauds is

quite uneven.

Note that a specific peculiarity of real-world FDS is the *delayed feedback* Dal Pozzolo et al. (2014), due to the fact that transaction labels are available several days later, once customers have reported unauthorized transactions. This is very detrimental in a concept-drifting environment since the supervised information is delayed by several days or weeks. In our experiments, the delay is modeled by a *gap* period (7 days) during which transactions are considered as non annotated yet (Figure 1). Indeed, the supervised information, frauds and genuine transactions, comes from two sources:

- the feedback provided by investigators. It is limited in number since only a few hundred transactions can be investigated per day.
- the delayed supervised transactions. It is the vast majority of the labels but they become available only after several days (e.g., one week, one month). After this delay, all unreported transactions are, in practice, considered genuine and can integrate the training dataset.

For more details about these two sources of feedback, see Dal Pozzolo et al. (2017).

Recent works include Abakarim et al. (2018), proposing a live credit card fraud detection system based on a deep neural network technology, specifically on on an auto-encoder, and comparing it with four different binary classification models. The proposed Deep NN Auto encoder has promising results in terms of F1 score.

### *3.1. How the plasticity/stability trade-off is set in practice ?*

After many duscussion with our industrial partner, it turns out that, for transactional companies, accuracy in terms of Pr@100 is the main objective, but this should not be pursued at the cost of low stability. High Pr@100 is the most important criterion to be considered for model selection yet, for similar Pr@100 performances, low forgetting should be preferred. In Section 6, it appears that some methods perform well according to one criterion but not the other.

| | Acronym | Setting | Strategy | $c$ classifier | Hyperparam. |
|---|---|---|---|---|---|
| Batch | Batch | batch (never) | Batch learning | 1 | - |
| Retrain | Retr | batch (1day) | Batch retraining | 1 | - |
| Incremental | Incr | CL (1day) | Baseline | 1 | - |
| Incremental (Ens) | IncrE | CL (1day) | Baseline | 10 | - |
| Experience Replay | ExpR | CL (1day) | Rehearsal, baseline | 1 | $r = .5, b = .01$ |
| Generative Replay | GenR | CL (1day) | Pseudo rehearsal | 1 | $r = .5, b = .01$ |
| Generative Replay (Ens) | GenRE | CL (1day) | Pseudo rehearsal | 10 | $r = .5, b = .01$ |
| Negative Correlation (Ens) | NCE | CL (1day) | Diversity | 10 | $\lambda = 1$ |
| Elastic Weight Consolidation | EWC | CL (1day) | Regularization | 1 | $\lambda = 0.01$ |
| Incremental Moment Matching | IMM | CL (1day) | Regularization | 1 | $\lambda = 0.1$ |
| Increm. Moment Matching (Ens) | IMME | CL (1day) | Regularization | 10 | $\lambda = 0.1$ |
| Frozen network | Frz | CL (1day) | Architectural | 1 | - |
| Frozen network (Ens) | FrzE | CL (1day) | Architectural | 10 | - |

Table 1: This table summarizes all the considered methods in this paper. More details can be found in Section 4. CL stands for continual learning, with the frequency of update in the brackets. Best values for the hyperparameters are reported. (Ens) indicates an ensemble version of the algorithm. The full considered parameters values are the following : $r = [.01, .05, .1, .2, .5, 1, 2]$ (the replay ratio), $b = [.01, .5, .99]$ (the generated proportion of frauds), $\lambda = 10^{[-6,-3,-2,-1,0,1,2,3,6]}$ (the regularization parameter value)

## 4. Continual learning for fraud detection

This section introduces a number of approaches (summarized in Table 1) to deal with continual learning in FDS. We consider some baselines and several state-of-the-art strategies which are relevant in the fraud detection context. Weight-transfer Lee et al. (2017) is the default transfer mechanism in our experiments: parameters for a new chunk are initialized with the parameters of the previous chunk. The initialization set (used for the first chunk) is composed of 61 days, taken just before the main data stream.

The objective is not to be exhaustive but to explore the potential of some approaches and define a sound strategy to assess specifically their stability/plasticity performance in a fraud detection setting. As an additional architectural strategy, we also test most of the following approaches in an ensemble setting.

In order to allow a statistically meaningful comparison between strategies all of them are implemented by adopting the same underlying classifier: a multi-

layered dense neural network, developed by the industrial partner and known to be effective in the FDS setting.

### *4.1. Baselines*

We consider three baseline approaches:

1. *Batch (Batch)*: the classifier is learned on the training initialization dataset and never updated.
2. *Retrain (Retr)*: a new classifier learned every day using the most recent available chunks (same size as initialization set).
3. *Incremental (Incr)*: the classifier is initialized as *Batch* and updated as new daily chunks arrive. No strategies are applied to tackle catastrophic forgetting.

### *4.2. Experience Replay (ExpR)*

Experience Replay implements the simplest rehearsal strategy, using a training dataset merging (uniformly sampled) past and recent transactions (current data chunk). The $r$ parameter is the ratio of old transactions with respect to new data. Note that such approach is not feasible in a FDS pipeline as it is not compliant with the GDPR requirements.

### *4.3. Generative Replay (GenR)*

This is the pseudo-rehearsal version of *ExpR* which uses a conditional generative adversarial neural network (CGAN, Mirza & Osindero (2014)) to generate past synthetic transactions Shin et al. (2017) and then comply with the GDPR issue. CGAN generation relies on two condition variables, representing the class (fraud/genuine) and the time of the transaction. The approach needs the setting of two hyperparameters; the proportion $b$ of frauds and the proportion $r$ of old transactions. The GAN is composed of a generator and a discriminator with the following structure: four layers of 256 nodes with leaky Relu activation functions. The input of the generator is compose of random noise of size 10 and the two condition variables.

### 4.4. Negative Correlation (NCE)

This is an incremental learning approach Liu & Yao (1999) aiming to encode the largest number of concepts in an ensemble of $M$ NN classifiers (see Brown et al. (2005a) for a review). The rationale consists in introducing a correlation penalty term in the error function of the $m$th NN in the ensemble

$$\mathcal{E}_m = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(F_m(n) - t(n))^2 + \frac{\lambda}{N}\sum_{n=1}^{N} p_m(n), \quad m = 1, \ldots, \tag{1}$$

where $F_m(n)$ is the output of the $m$th NN for the $n$th sample, $t(n)$ is the $n$th target value. The first term is the empirical loss function of the $m$th NN while the second term

$$p_m(n) = (F_m(n) - F(n)) \sum_{m' \neq m} (F_{m'}(n) - F(n)) \tag{2}$$

is the correlation penalty function, with $F(n)$ being the ensemble output. This term accounts for the diversity of the classifiers and is controlled by the parameter $\lambda$ Brown et al. (2005b).

### 4.5. Elastic Weight Consolidation (EWC)

This is a regularization strategy which aims to constrain the variability of the NN parameters over the learning horizon Kirkpatrick et al. (2017) by adding a term to the loss function:

$$\mathcal{L}_{EWC}(w_t) = \mathcal{L}(w_t) + \sum_i \frac{\lambda}{2} F_i (w_{t,i} - w_{t-1,i})^2 \tag{3}$$

where $\mathcal{L}_{EWC}(w_t)$ is the EWC loss, $w_{t,i}$ is the $i$th weight after the $t$ chunk, $\mathcal{L}(w_t)$ is a well-suited loss function for the problem at hand (here, classification). Notice our EWC implementation uses chunk-specific importance parameters. Online versions of EWC Chaudhry et al. (2018); Schwarz et al. (2018) exist but were not considered in this work. $\lambda$ (to be tuned) sets how important the L2 constraint on the parameters is and $F_i$ is the inverse of the importance indicated by the Fisher information matrix. The parameters associated with higher Fisher diagonal elements are considered to be more important for the previous tasks

and are therefore less flexible Seff et al. (2017). In practice, the computation of this matrix is quite expensive in our case. For that reason, the ensemble version of *EWC* was not considered.

### *4.6. Incremental Moment Matching (IMM)*

In IMM Lee et al. (2017), Gaussian distributions are used to approximate the posterior distribution of the NN weights. Mean-IMM reduces to

$$\mu_{t+1} = \sum_{j=1}^{t} \alpha_j \mu_j \tag{4}$$

where $\mu_t$ is the mean value (and optimal weigth value) for a given NN weight after $t$ chunks, and $\alpha_j$ are weighting factors that we set to $1/t$. Therefore, Each NN weight takes the new and old $\mu_j$ into account.

Mode-IMM is a variant where the covariance information of the posterior of Gaussian distributions is used. Also, the authors suggest three transfer parameter mechanisms: (i) weight-transfer, L2-transfer (similar to the procedure of EWC, but without the Fisher Matrix), and Drop-transfer ($\mu_t$ is used to initialize the dropout procedure). All in all, there are six variants for this algorithm.

We tried the mean/weight-transfer and mean/L2-transfer. The latter gave better results, so we did not report the former for the sake of simplicity. The parameter to tune is $\lambda$ (from the L2-transfer part): the lower $\lambda$, the more the weights are free to evolve.

### *4.7. Frozen network (Frz)*

This approach (also known as "fine-tuning" in NLP Howard & Ruder (2018)) consists of learning a multi-layer neural network on a large quantity of training data (in our case, the initialization set) and then making only the last layer trainable. The idea is that a large part of the past behavior will therefore be preserved. For example, the *Copy Weight with Re-init* (CWR) algorithm Lomonaco & Maltoni (2017) uses this strategy with a re-initialization at each new data chunk. In this work, we did not use re-initialization: chunks are relatively small so we can benefit from transferring last weights from previous chunks.

## 5. The plasticity/stability matrix

This section introduces the *plasticity/stability matrix*, an original visualization technique, adapted for fraud detection, which represents in a compact way the accuracy of a continual learning algorithm in terms of its plasticity and stability components. The rationale consists in constructing a square matrix $M(t_i, t_j)$ where the $(t_i, t_j)$ term denotes the accuracy of a model trained up to day $t_j$ and tested on the day $t_i$. Figure 2.a. shows the plasticity/stability matrix for two methods over 85 days. Note that the lower (upper) triangle shows the accuracy of a model trained up to a day $t$ and tested on a day $> t$ ($< t$). This implies that the main diagonal (upper triangular) part summarizes the plasticity (stability) properties of the model. In other words, the better the accuracy in the main diagonal, the higher is the capability of the model to fast on recent concepts (plasticity). The better the accuracy in the higher triangular part, the higher is the resilience (stability) of the model to catastrophic forgetting.

In Figure 2.c. and 2.d., three portions can be identified : (i) this portion summarizes how the model is able to anticipate future distant drifts in the data distribution. This task is the harder the lower we go (i.e. the larger is the difference between the last training day and the test day) and it is not realistic to expect high accuracy in that region. For this reason we discard it in our final assessment. (ii) this portion between the main diagonal and the 7-th diagonal (which is the gap duration in days) is a more appropriate measure of the plasticity in the delayed feedback setting. This part refers indeed to the performance of the algorithm during the gap interval. (iii) The top triangle above the 7-th diagonal can be referred as the *catastrophic triangle* since it can be used to quantify to what extent the updating classifier is forgetting. We can also use the 7-th diagonal as a proxy of an ideal model, as the model was just updated with the labels he is supposed to predict.

In Fig 2.b., we plot the accuracy values of the same row from the catastrophic triangles of two considered methods. This comparison shows a very different behavior in terms of catastrophic forgetting. The first method (in red)

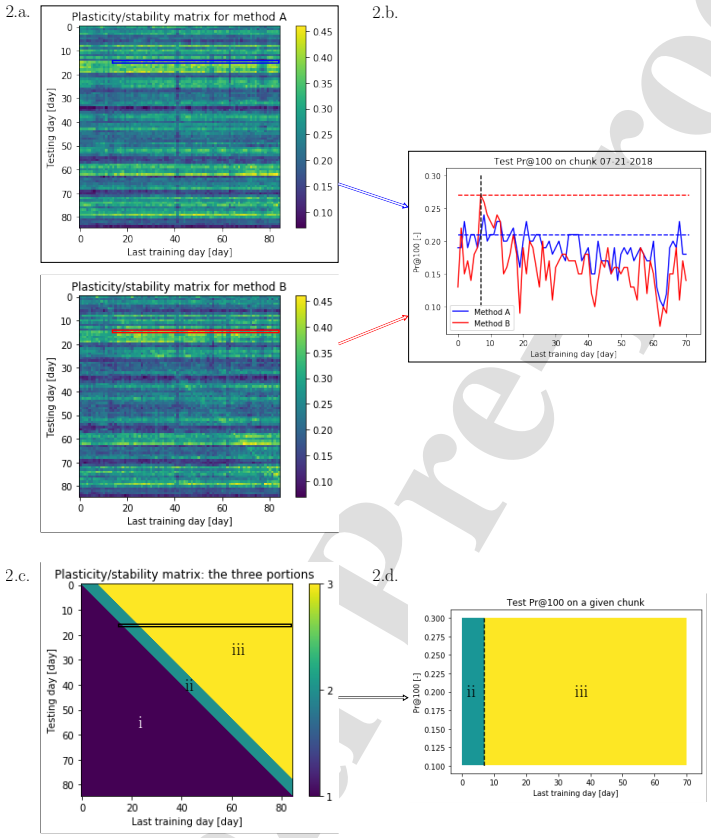


Figure 2: By comparing the rows of the upper triangular matrices (see 2.a), we compare the forgetting behavior for each chunk. The catastrophic triangle (portion (iii) on 2.c and 2.d) is obtained by considering pairs of model updates and testing days. It also takes the gap (portion (ii) on 2.c and 2.d) and temporal behavior into account. The ideal behavior is rows with constant accuracy in all the part (iii) of the matrix. The contrast could be enhanced using calibrated accuracy measures Siblini et al. (2020). 2.b.: Horizontal dashed lines represent an optimal behavior, for a classifier with no forgetting at all.

experiences a high forgetting rate made explicit by the deterioration of the accuracy the farther we go with respect to the peak. This is not the case of the blue method characterized by a more stable accuracy.

To quantify the degree of catastrophic forgetting of a generic model, we define a regret term

$$\mathcal{R}(t_i, t_j) = -[M(t_i, t_j) - M(t_i, t_i + l_g)]/M(t_i, t_i + l_g), t_j > t_i \tag{5}$$

where $M(t_i, t_j)$ is the $t_i, t_j$ entry of the plasticity/stability matrix and $l_g$ is the length of the gap (in days). Remember $M(t_i, t_j)$ means the model was trained until day $t_j$, saw labels until day $t_j - l_g$, and is evaluated on $t_i$. $M(t_i, t_i + l_g)$ is the optimal accuracy for $t_i$ since the labeled chunk corresponding to day $t_i$ just becomes available.

It is then possible to run a Friedman/Nemenyi test Demsar (2006) to find the method with the lowest regret distribution. Note that the plasticity/stability matrix is not then only a qualitative way to illustrate the catastrophic forgetting but it allows the implementation of a statistically sound procedure to compare two methods in terms of their plasticity/stability properties.

## 6. Experimental assessment

This section details the experimental session in terms of streaming assessment procedure, dataset characteristics, implemented code and quantifies the plasticity and stability performance using the methodology discussed in the previous section.

### *6.1. The assessment setting*

The pseudo-code (Algorithm 1) sketches the assessment procedure used to compare the different approaches. Given a continual learner $cl$, once a new daily chunk is available at time $t$, the procedure updates the online model, predicts the fraud probability and computes the accuracy. Keep in mind that the data from the gap set are considered as not labeled (Section 2). We refer the reader

**Algorithm 1:** The streaming assessment procedure

Initialize $cl$ using classical batch learning ($I = 61$ chunks);
$l_g \leftarrow$ length of the gap set (in days);
**for** $t = (I + l_g) : T$ **do**
    % update the model, ignoring the gap set
    $D_{t-l_g} \leftarrow$ new training chunk (one day);
    Update $cl$ with $D_{t-l_g}$ using backpropagation;
    % compute accuracy for this day
    $D_t \leftarrow$ test chunk (one day);
    Rank all transactions of $D_t$ according to $cl$ risk scores;
    Update Pr@100 and AUPRC statistics;
**end**

to Lebichot et al. (2017) for semi-supervised strategies which allow to exploit the gap set in real-time.

### *6.2. Data and code*

The experimental dataset is an extract from a real e-commerce transaction stream provided by the industrial partner. It is made of 50M e-commerce transactions over 153 days (61 for models initialization, 7 gap days, 15 validation days, to set the hyperparameters, and 70 test days). Validation days are used to tune the hyperparameters detailed in Table 1. Each experiment is replicated five times with different random seeds, and the average and std are reported. The fraud ratio is 0.201% and each transaction is described by 23 features. Though data cannot be made available for confidential reasons, a public, short, anonymized version of the data can be found here Machine Learning Group - ULB. The data collections is similar to our previous work Dal Pozzolo et al. (2014). The dataset was just updated with more recent transactions. All experiments were carried out on a server with 10 cores, 256 GB RAM, and an Asus GTX 1080 TI. Keras Chollet et al. (2015), and code from the original EWC paper Kirkpatrick et al. (2017) was used for the NNs implementation and

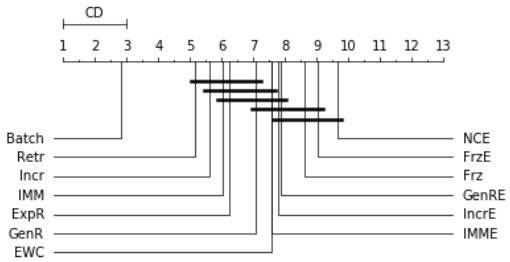


Figure 3: Friedman-Nemenyi test based on the card-based Pr@100 (70 days with one score per day, the large the better). The plot compares various approaches and baseline approaches. A method is considered as significantly better than another if its mean rank is more than the critical difference $CD$ higher (the higher, the better). Each of the 70 results is averaged on five runs. The averages over the 5x70 individual results are reported on Table 2.

training. The code is not made publicly available as the content would disclose some of the industrial secrets of our private partner. Overall times provided in Table 1 give an idea of the relative time-intensive steps of the 13 methods: Ensemble versions are longer to obtain, *Retrain* is quite slow compare tho the rest, and *EWC*'s Fisher Matrix is long to obtain (faster estimates exists, but we did not implement them).

### *6.3. Overall accuracy assessment*

Table 2 presents the results on the main diagonal of the plasticity/stability matrix in terms of accuracy and processing time (to process the full stream). Figure 3 and Figure 4 present the results under the form of a Friedman-Nemenyi test, for the Pr@100 and AUPRC metric, respectively.

First of all, the results appear to be coherent from the two metrics perspective. If we take the best method and the ones which are significantly the closest, we obtain a set of three methods: *NCE*, *FrzE*, *IncrE*. The most accurate strategies in terms of Pr@100 and AUPRC are therefore *Diversity*, *Architectural*, and

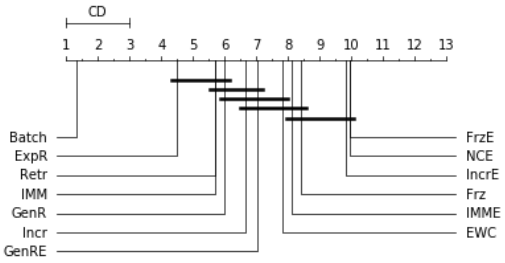


Figure 4: Friedman-Nemenyi test based on the card-based AUPRC. See Figure 3 for more details.

| | Mean Pr@100 | Std Pr@100 | Mean AUPRC | Std AUPRC | Overall time |
|---|---|---|---|---|---|
| Batch | 20.54 % | 5.95 % | 6.97 % | 2.36 % | 0 s* |
| Retrain | 22.64 % | 6.17 % | 10.11 % | 2.83 % | 182 min |
| Incremental | 23.04 % | 6.15 % | 10.31 % | 2.85 % | 19 min |
| Incremental (Ens) | 24.28 % | 6.83 % | 11.20 % | 3.20 % | 106 min |
| Experience Replay | 23.38 % | 6.65 % | 9.54 % | 2.79 % | 20 min |
| Generative Replay | 24.03 % | 7.23 % | 10.16 % | 3.00 % | 23 min |
| Generative Replay (Ens) | 24.32 % | 7.11 % | 10.05 % | 3.01 % | 71 min |
| Negative correlation (Ens) | **26.24 %** | **7.96** % | **11.66 %** | **3.69** % | 126 min |
| Elastic Weight Consolidation | 24.19 % | 7.00 % | 10.59 % | 3.05 % | 657 min |
| Incremental Moment Matching | 23.15 % | 6.78 % | 9.90 % | 2.89 % | 31 min |
| Increm. Moment Matching (Ens) | 24.02 % | 7.25 % | 10.72 % | 3.00 % | 249 min |
| Frozen network | 25.03 % | 6.43 % | 10.92 % | 2.87 % | 18 min |
| Frozen network (Ens) | 25.36 % | 6.49 % | 11.28 % | 2.90 % | 89 min |

Table 2: This table summarizes the results in terms of AUPRC and Pr@100. Overall time is an indicative execution time to process the whole data stream (3 months of data). Each result is averaged on five runs. Notice that since the NN training is performed on GPU and data manipulation on CPU, the training part is not necessarily the most time-consuming part of the update. * This model is never updated.

*Ensemble*. *GenRE* (*IMME*) is also significantly equivalent to the top-3 if we limit to consider Pr@100 (AUPRC). Also, As observed in Lebichot et al. (2020), batch approaches are sub-optimal. This supports the hypothesis of the presence of concept drift and the necessity for plasticity mechanism.

We see that superior accuracy can be obtained in various ways (as it is often the case in real case studies). (i) the ensemble version of the algorithms outperforms their single counterpart (with little surprise). (ii) *NCE* does not address catastrophic forgetting but instead optimizes diversity among its ensemble. It means that diversity can be as important as minimizing forgetting. (iii) *FrzE* (and *Frz*) takes advantage of a fixed layer, trained on the (large) initialization dataset, and a trainable layer which allows to adapt to the transactions stream. After investigation, we found that some chunks can mislead the baseline model persistently. With *FrzE*, the untrainable part is unaffected and the trainable part can be corrected more quickly afterwards.

### *6.4. Stability assessment*

On the basis of the regret terms $\mathcal{R}(t_i, t_j)$ (Eq. 5), we may quantify the degree of catastrophic forgetting of the benchmarked methods. We perform a Friedman/Nemenyi test Demsar (2006) on all $\mathcal{R}(i, j)$ corresponding to the catastrophic triangle (portion (iii)). Figure 6 and 7 reports the results in terms of Pr@100 and PRAUC, respectively.

From these two figures, methods can be discussed, per decreasing order :

- *IMME* and *IMM* comes among the firsts. Because they keep track of all the weights for all previous chunks, the model is strongly stabilized but at the end of the dataset, it turns detrimental: regrets are better at the beginning of the stream than at the end.
- *ExpR* explicitly replays old transactions, so its good ranking is not a surprise. In addition *GenR* and *GenRE* performs similarly, which shows that our GAN is able to replicate realistic transactions.
- *FrzE* and *Frz* are also quite stable for the reason we discuss in Section 6.3.

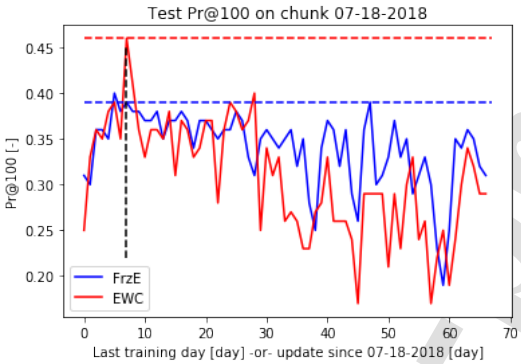


Figure 5: Typical behavior inside a row of the catastrophic triangle (portion (iii)). The plot shows how the successive update of the model affect its ability to classify the transactions of chunk 07-18-2018. The ideal behavior is suggested by the dashed lines. In this illustration, *EWC* exhibits a large forgetting.

- *NCE* performs less efficiently: it is not surprising as it does not focus on addressing catastrophic forgetting: it manages diversity instead.
- *Incr* and *IncrE* have no strategies to alleviate the forgetting. They perform quite poorly.
- *EWC* has the largest performance peaks after the gap (see Figure 5), and the accuracy strongly decreases afterward. These two drawbacks lead to large regrets.

We can also conclude that the ensembles help to increase the accuracy, but do not change the intrinsic behavior regarding catastrophic forgetting, as methods are close to their ensemble counterpart.

### *6.5. Take-home message*

Given our main objectives (recall Section 3), we reached the following conclusions, after discussing with the industrial domain experts:

- High Pr@100 is the most important criterion to be considered for model selection yet, for similar Pr@100 performances, low forgetting should be preferred.
- A method without any anti-forgetting mechanism can be efficient in terms of accuracy, but highly inadequate in terms of stability: *Incr*, *IncrE*, *NCE*.
- Methods with anti-forgetting mechanisms, like *ExpR*, *GenR*, *IMME*, and *Frz*, may trade adaptability for stability and consequently can have low accuracy.
- By changing their hyperparameters, some methods can passe from a high plasticity/low stability setting to the symmetric situation. For example, *EWC* was initially designed to avoid forgetting (by increasing stability but lower adaptability) so it should be better than *Incr* on stability and worse on plasticity. But we observe the opposite behavior in this work because the primary objective is Pr@100 and hyperparameters tuning was performed in this sense.

Overall, we reached the conclusion that the best trade-off is attained by the adoption of *FrzE*. This conclusion is only true for our particular plasticity/stability trade-off. Of course, another trade-off (for example maximizing stability) can lead to other hyperparameters, and therefore to other conclusions.

## 7. Conclusion and Future Work

The paper assesses the plasticity/stability trade-off in the context of continual credit card fraud detection. The experimental assessment does not limit to consider which methods are the most accurate but aims to return a quantitative measure of their degree of catastrophic forgetting. This is made possible thanks to an original procedure to visualize and quantify the catastrophic forgetting in data streams with delayed feedback.

We also discuss the fact that addressing catastrophic forgetting is a trade-off between plasticity and stability. Having quantify both, we include and discuss

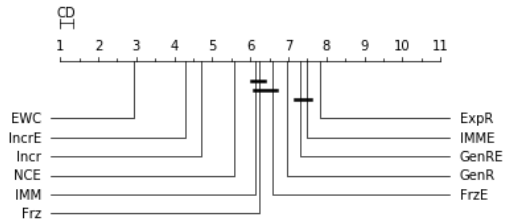


Figure 6: Friedman-Nemenyi test based on the regrets built on the card-based Pr@100 from the catastrophic triangle (portion (iii)). The larger the better. Acronyms are reported on Table 1.

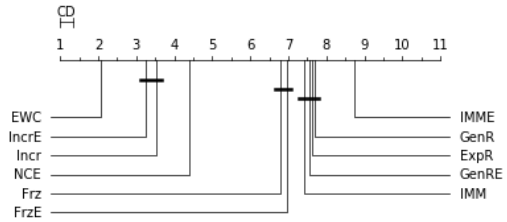


Figure 7: Friedman-Nemenyi test based on the regrets built on the card-based PRAUC. See Figure 6 for more details.

the best trade-off, in accordance with field experts. Overall, the best method is *FrzE*, an ensemble of frozen neural networks because it performs well in terms of Pr@100, PRAUC, and achieve satisfactory reduced forgetting.

We believe our conclusions may be broadened to other single incremental task learning applications or Fintech domains. For example, the plasticity/stability matrix is also relevant when no delayed feedback is present. On the basis of this study, the industrial partner is now considering continual learning for production FDS.

Further work will include testing the catastrophic triangle on longer streams, to see if recurrent patterns (e.g. Christmas e-shopping) are effectively preserved by the methods, and more investigations on the accuracy and the properties of the generative methods. A last research direction will be to explore different plasticity/stability trade-offs.

**Acknowledgements**

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. B. Lebichot also thanks the LouRIM, Université catholique de Louvain, Belgium for its support.

**References**

Abakarim, Y., Lahby, M., & Attioui, A. (2018). An efficient real time model for credit card fraud detection based on deep learning. In *Proceedings of the 12th international conference on intelligent systems: theories and applications* (pp. 1–7).

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)* (pp. 1351–1355). IEEE.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005a). Diversity creation methods: a survey and categorisation. *Information Fusion*, *6*, 5–20.

Brown, G., Wyatt, J. L., & Tiňo, P. (2005b). Managing diversity in regression ensembles. *Journal of machine learning research*, *6*, 1621–1650.

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). Scarff: a scalable framework for streaming credit card fraud detection with spark. *Information fusion*, *41*, 182–194.

Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, .

Chaudhry, A., Dokania, P. K., Ajanthan, T., & Torr, P. H. (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 532–547).

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Cichon, J., & Gan, W.-B. (2015). Branch-specific dendritic ca 2+ spikes cause persistent synaptic plasticity. *Nature*, *520*, 180–185.

Cossu, A., Carta, A., Lomonaco, V., & Bacciu, D. (2021). Continual learning for recurrent neural networks: an empirical evaluation. *Neural Networks*, *143*, 607–627.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). Credit card fraud detection and concept-drift adaptation with delayed supervised information. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 1–8). IEEE.

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 transactions on neural networks and learning systems*, *29*, 3784–3797.

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert System with Applications*, *10*, 4915–4928.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233–240).

Demsar, J. (2006). Statistical comparaison of classifiers over multiple data sets. *Journal of Machine Learning Research 7*, (pp. 1–30).

Ditzler, G., & Polikar, R. (2012). Incremental learning of concept drift from streaming imbalanced data. *ieee transactions on knowledge and data engineering*, *25*, 2283–2301.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, *3*, 128–135.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, .

Hayes, T. L., Cahill, N. D., & Kanan, C. (2019). Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 9769–9776). IEEE.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, .

Kemker, R., McClure, M., Abitino, A., Hayes, T. L., & Kanan, C. (2018). Measuring catastrophic forgetting in neural networks. In *Thirty-second AAAI conference on artificial intelligence*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A. et al.

(2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, *114*, 3521–3526.

Kou, Y., Lu, C.-T., Sirwongwattana, S., & Huang, Y.-P. (2004). Survey of fraud detection techniques. In *IEEE International Conference on Networking, Sensing and Control, 2004* (pp. 749–754). IEEE volume 2.

Lebichot, B., Braun, F., Caelen, O., & Saerens, M. (2017). A graph-based, semi-supervised, credit card fraud detection system. In *Complex Networks & Their Applications V: Proceedings of the 5th International Workshop on Complex Networks and their Applications (COMPLEX NETWORKS 2016)* (pp. 721–733). Cham: Springer International Publishing.

Lebichot, B., Paldino, G. M., Bontempi, G., Siblini, W., He-Guelton, L., & Oblé, F. (2020). Incremental learning strategies for credit cards fraud detection. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 785–786). IEEE.

Lee, S.-W., Kim, J.-H., Jun, J., Ha, J.-W., & Zhang, B.-T. (2017). Overcoming catastrophic forgetting by incremental moment matching. In *Advances in neural information processing systems* (pp. 4652–4662).

Li, H., Krishnan, A., Wu, J., Kolouri, S., Pilly, P. K., & Braverman, V. (2021). Lifelong learning with sketched structural regularization. In *Asian Conference on Machine Learning* (pp. 985–1000). PMLR.

Li, Z., & Hoiem, D. (2017). Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, *40*, 2935–2947.

Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. *Neural networks*, *12*, 1399–1404.

Lomonaco, V., & Maltoni, D. (2017). Core50: a new dataset and benchmark for continuous object recognition. *arXiv preprint arXiv:1705.03550*, .

Losing, V., Hammer, B., & Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, *275*, 1261–1274.

Machine Learning Group - ULB (). Credit card fraud detection (consulted on 2020-06-28).

Maltoni, D., & Lomonaco, V. (2019). Continuous learning in single-incremental-task scenarios. *Neural Networks*, *116*, 56–73.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation* (pp. 109–165). Elsevier volume 24.

Mermillod, M., Bugaiska, A., & Bonin, P. (2013). The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, *4*, 504.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, .

Pang, G., Shen, C., Cao, L., & Hengel, A. v. d. (2020). Deep learning for anomaly detection: A review. *arXiv preprint arXiv:2007.02500*, .

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, .

Polikar, R., Upda, L., Upda, S. S., & Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, *31*, 497–508.

Ramasesh, V. V., Lewkowycz, A., & Dyer, E. (2021). Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, *97*, 285.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 2001–2010).

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., & Tesauro, G. (2018). Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, .

Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, *7*, 123–146.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*, .

Sahoo, D., Pham, Q., Lu, J., & Hoi, S. C. (2018). Online deep learning: learning deep neural networks on the fly. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 2660–2666).

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*, e0118432.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., & Hadsell, R. (2018). Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning* (pp. 4528–4537). PMLR.

Seff, A., Beatson, A., Suo, D., & Liu, H. (2017). Continual learning in generative adversarial nets. *arXiv preprint arXiv:1705.08395*, .

Shin, H., Lee, J. K., Kim, J., & Kim, J. (2017). Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems* (pp. 2990–2999).

Siblini, W., Fréry, J., He-Guelton, L., Oblé, F., & Wang, Y.-Q. (2020). Master your metrics with calibration. In *International Symposium on Intelligent Data Analysis* (pp. 457–469). Springer.

Sun, Y., Tang, K., Zhu, Z., & Yao, X. (2018). Concept drift adaptation by exploiting historical knowledge. *IEEE transactions on neural networks and learning systems*, *29*, 4822–4832.

van de Ven, G. M., Siegelmann, H. T., & Tolias, A. S. (2020). Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, *11*, 1–14.

Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data mining and knowledge discovery*, *18*, 30–55.

Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3987–3995). JMLR. org.

B. Lebichot Orcid: 0000-0003-2188-0118
W. Siblini Orcid: 0000-0002-4193-2061
G.M. Paldino Orcid: 0000-0002-8680-9403
Y.-A. Le Borgne Orcid: 0000-0001-5679-7758
F. Oblé
G. Bontempi Orcid: 0000-0001-8621-316X

B. Lebichot: Conceptualization, Methodology, Software, Writing - Original draft preparation
W. Siblini: Conceptualization, Methodology, Software
G.M. Paldino: Writing - Reviewing and Editing
Y.-A. Le Borgne: Conceptualization, Methodology
F. Oblé: Conceptualization, Supervision
G. Bontempi: Conceptualization, Supervision

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: