International Journal on Document Analysis and Recognition (IJDAR)
https://doi.org/10.1007/s10032-024-00467-w

ORIGINAL PAPER

Experimental study of rehearsal-based incremental classification of document streams

Usman Malik[1] · Muriel Visani[1,2] · Nicolas Sidere[1] · Mickael Coustaty[1] · Aurelie Joseph[3]

Received: 20 December 2022 / Revised: 19 November 2023 / Accepted: 15 April 2024

Abstract
This research work proposes a novel protocol for rehearsal-based incremental learning models for the classification of business document streams using deep learning and, in particular, transformer-based natural language processing techniques. When implementing a rehearsal-based incremental classification model, the questions raised most often for parameterizing the model relate to the number of instances from "old" classes (learned in previous training iterations) which need to be kept in memory and the optimal number of new classes to be learned at each iteration. In this paper, we propose an incremental learning protocol that involves training incremental models using a weight-sharing strategy between transformer model layers across incremental training iterations. We provide a thorough experimental study that enables us to determine optimal ranges for various parameters in the context of incremental classification of business document streams. We also study the effect of the order in which the classes are presented to the model for learning and the effects of class imbalance on the model's performances. Our results reveal no significant difference in the performances of our incrementally trained model and its statically trained counterpart after all training iterations (especially when, in the presence of class imbalance, the most represented classes are learned first). In addition, our proposed approach shows an improvement of 1.55% and 3.66% over a baseline model on two business documents dataset. Based on this experimental study, we provide a list of recommendations for researchers and developers for training rehearsal-based incremental classification models for business document streams. Our protocol can be further re-used for other final applications.

Keywords Document classification · Natural language processing · Deep learning · Incremental learning · Transformers · Rehearsal-based incremental learning

1 Introduction

A major part of human communication, formal or informal, takes place through documents. The need for document processing and analysis is especially crucial in the corporate sector, where various organizational decisions depend upon the information extracted from business documents (letters, invoices, quotations, tax notices, resumes, bank statements, etc). Some of the most usual document processing tasks needed by corporate organizations include document classification, clustering, forensics, and information extraction.

Document classification refers to automatically identifying and assigning the correct category for a given document, based on clues hidden in the document's content [1]. This contextual information used for classification can be in the form of text, images, or both.

Recent studies show that deep learning techniques can be used to perform different types of document processing

✉ Usman Malik
usmanmalik57@gmail.com

✉ Muriel Visani
muriel.visani@univ-lr.fr

Nicolas Sidere
nicolas.sidere@univ-lr.fr

Mickael Coustaty
mickael.coustaty@univ-lr.fr

Aurelie Joseph
aurelie.joseph@getyooz.com

[1] La Rochelle Université, Laboratoire Informatique, Image et Interaction (L3i), 17042 La Rochelle, France

[2] French Military Center for Epidemiology and Public Health (CESPA), 13014 Marseille, France

[3] Yooz, 1 Rue Fleming, 17000 La Rochelle, France

Published online: 11 May 2024

and analysis tasks [2, 3]. But, while traditional deep learning algorithms rely on static training and test sets, most real-life datasets for business documents classification are constantly evolving. Indeed, every day, companies receive/digitize new documents which can belong to existing categories or represent new document categories. Traditional deep learning approaches, which learn from static data, are not optimal in such cases, as most of them would have to be retrained from scratch every time a new class (or at least significant information) is added, resulting in excessive time and resource consumption. Incremental learning models are thus more suitable for most real-life document classification applications.

Incremental learning is a branch of machine learning where models are trained on the go with the arrival of new data during training [4, 5]. The model, after being trained on the initial dataset, updates itself to adjust to new data distribution at each iteration where new data is added.

In this paper, we focus on incremental business document classification, which comprises classifying automatically all inbound communication, i.e. diverse document streams, including emails, invoices, resumes, tax notices, bank statements, etc. [6]. More precisely, in this research work, we propose a rehearsal-based incremental learning protocol for the classification of business documents based on the text automatically extracted from these documents. The idea behind rehearsal strategies is that, along with the new data arriving on the go, a subset of data from previous training iterations is also kept in memory and used for training during later iterations [5, 7], to avoid "catastrophic forgetting" for the classes learned during early learning iterations [8].

We experiment with the proposed protocol in the presence of documents written in two different languages: English and French, and in the presence of balanced, as well as highly imbalanced, datasets.

The three main contributions of this paper are:

- We introduce a novel protocol for incremental document classification and perform an incremental classification of document streams using weight sharing strategy between transformer models layers across multiple incremental iterations, which have not been explored before, to the best of our knowledge.
- Using different datasets, we extensively compare its performances with its static counterpart and a baseline approach and investigate the effects of class imbalance on both models.
- We provide recommendations for setting the main parameters required for rehearsal-based incremental document classification models: (a) the number of instances from "old" classes (classes learned in previous training iterations) that need to be kept in memory at each iteration in order to avoid catastrophic forgetting, (b) the optimal number of new classes to add at each iteration, and (c) the effect on the overall performance of the order in which the classes are presented to the model for learning. We formulate these recommendations based on an extensive experimental study.

This article is divided into six sections. Section 2 sheds light on some of the existing works for document classification and incremental learning. The proposed approach, methodology, and the details of the experiments performed in this research work are presented in Sect. 3. Section 4 presents the analysis of our results, while Sect. 5 contains a discussion and recommendations. Finally, Sect. 6 concludes the paper and presents future directions for this research.

2 Literature review

Several researchers have proposed models for business document classification using textual, image, and multimodal information, possibly using incremental learning.

Since this paper lies at the crossroad of document classification and incremental learning, this section is further divided into three sub-sections. While Sect. 2.1 briefly reviews recent works for document classification, Sect. 2.2 presents a thorough literature review for incremental learning, and Sect. 2.3 discusses the choices adopted in this research work.

2.1 Document classification

Typical business document classification workflows include document capture, image analysis, Optical Character Recognition (OCR) for recognizing the text from bitmap images, text analysis, assigning the appropriate category to the document, and document routing to some business process based on the category assigned. However, the complexity and diversity of informative elements, backgrounds, and geometric layouts make it difficult to achieve very good results for automatic document classification [9].

Asim et al. [1] present a two-staged text classification system (TSCNN). The first stage includes using a filter-based feature selection method - Normalized Difference Measure (NDM) - to eliminate redundant or irrelevant features. These fine-tuned features are then fed to multichannel Convolutional Neural Networks (CNN) for classifying the input document into the relevant category. Using two publicly available datasets (BBC News and 20 News-Group), this approach achieves an accuracy of 99.251% and 91.746%, respectively. While it outperforms the baselines, it requires an additional feature engineering step, which (for large datasets) may result in increased training time compared to contemporary models.

Alhaj et al. [10] propose using a stemming technique to reduce the high dimensionality of feature vectors and save computational cost. Using three stemming methods (Information Science Research Institute, Tashaphyne, and ARLStem) and three machine learning algorithms (naive bayes, support vector machines, and K-nearest neighbours), they classified Arabic text documents. The best results (94.64% Micro-F1) are obtained using ARLStem for dimensionality reduction, combined with support vector machines. The approach was not tested on other mainstream languages besides Arabic.

d'Andecy et al. [9] compare the performance of CNN-RNN based approach and a custom incremental learning approach for automatic document classification using the OCR based textual representation of documents from Digital Mailroom dataset. They reported that CNN-RNN based approach outperformed the Incremental classification approach by achieving an accuracy of 94%. Though the results achieved via custom incremental learning approach serve as a proof of concept for the viability of the approach, the performance achieved is less than the state of art performance achieved via CNN-RNN based approach.

Shahkolaei et al. [11] use log-Gabor filter for text/non-text image segmentation and then SVM for classification. On two publicly available datasets (Visual document image quality assessment and MHDID), an accuracy of respectively 76.11% and 85.07% was achieved. This model, however, is only tested with Arabic language documents, and the extent of model adaptability to other languages is not known.

Some other interesting rule-based and machine learning approaches for document classification are proposed in [9, 12–14]. A comparison of some of the existing approaches for document classification is presented in Table 1.

2.2 Incremental learning

As explained in Sect. 1, incremental learning (IL) is desirable for modern document classification systems because it allows for efficient resource utilization by not having to retrain the system from scratch when new documents/information arrive. It reduces memory consumption by avoiding or limiting the huge quantity of data that must be stored for the proper functioning of the system. Incremental learning most closely resembles human learning [15].

In this research, we employ deep learning for incremental document classification. The choice of deep learning is motivated by its outstanding performances and its minimal requirements for human supervision.

Luo et al. [16] classified incremental learning scenarios into three categories: instance incremental learning, class incremental learning, and instance and class incremental learning. Instance incremental learning keeps the number of classes fixed while the number of instances per class expands during each incremental learning stage. In class incremental learning, new classes are added only during each incremental learning stage. Finally, in instance and class incremental learning, the number of instances from existing classes expands along with the addition of instances from new classes. In the rest of this paper, we follow this convention to refer to IL scenarios (instance incremental learning, class incremental learning, and instance and class incremental learning). We use the term "task incremental learning" (learning new tasking within the same domain [17]) to refer collectively to these three IL scenarios.

The major risk associated with incremental learning is "catastrophic forgetting" (CF): a sharp performance decline for already learned tasks after the acquisition of new data [8]. Catastrophic forgetting can be caused by several phenomena, including activation drift, weight drift, task-recency bias (referring to the bias towards the most recently-learned tasks), and inter-task confusion [15]. Several IL techniques have been proposed to reduce the risk of catastrophic forgetting while simultaneously learning new information.

But, trying to prevent catastrophic forgetting can cause another major issue: intransigence, or the unwillingness to learn new tasks [18]. Effective IL systems must find a fine balance between catastrophic forgetting and intransigence, a tradeoff called the stability-plasticity dilemma [19].

In the next section, we provide a thorough review of the state-of-the-art IL techniques aiming at finding a good trade-off between stability and plasticity. To this end, on the basis of recent works, four categories of incremental learning strategies are discussed in the next four sub-sections: regularization approaches, architectural strategies, variational continual learning (VCL), and rehearsal-based approaches.

2.2.1 Regularization approaches

Regularization approaches mitigate catastrophic forgetting by introducing a special regularization term to classification loss. Based on recent studies, regularization approaches can be divided into two major categories: weight regularization strategies and distillation strategies.

Weight Regularization Strategies Weight regularization strategies are based on the hypothesis that the previous information learned by a neural network can be preserved by (i) assessing the importance of weights relevant to learn previous information and (ii) restricting the learning rate for new information. An additional loss is incorporated in the loss function (in addition to the cross-entropy loss), which can be formulated as:

$$L_{reg}(\theta^t) = \frac{1}{2} \sum_{i=1}^{|\theta^{t-1}|} \Omega_i (\theta_i^{t-1} - \theta_i^t)^2$$

Table 1 A summary of existing works for document classification

Existing work	Document type	Methodlogy	Advantages	Limitations
Asim et al. [1]	Health & security	Normalized difference measure and CNN	High accuracy and robust feature selection using NDM	Increased training time and limited scalability
d'Andecy et al. [9]	Digital mailroom documents	Custom incremental learning approach	Shows viability of incremental learning	Approach doesn't achieve state-of-the-art results
Sanchez-Pi et al. [12]	BBC news and 20 news-group	Ontologies	High explainability	Low generalization, requires creating ontology for every domain
Gayathri et al. [13]	Health	Ontologies & semantic document description	High explainability	High domain dependence and low generalization
Walczak et al. [14]	Research articles	Keyword identification through text analytic	Simpler implementation and high explainability	High domain dependence and low generalization
Alhaj et al. [10]	CNN Arabic corpus	Stemming for preprocessing and & traditional ML algos e.g. SVM, KNN etc	Reduced number of features and faster training	Approach only tested on Arabic texts, requires additional preprocessing step
Shahkolaei et al. [11]	Visual document image & MHDID datasets	Log-gabor filter for preprocessing and SVM	Reduced number of features and faster training	Approach only tested with the Arabic language

where θ_i^t is weight i of the network trained for the current task t, θ_i^{t-1} is the value of weight parameter i at the end of training on task $t-1$, $|\theta^{t-1}|$ stands for the total number of weights in the network, and Ω_i contains importance values for all the network weights.

Zeng et al. [20] propose the orthogonal weights modification (OWM) approach for incremental image classification. By restricting the direction for weights updates of each parameter, it protects previously gained knowledge.

Farajtabar et al. [21] propose the orthogonal gradient descent (OGD) approach which, when a new task arrives, determines the orthogonal basis S of the previous task and then transforms the current task's original gradient to a new gradient perpendicular to S.

Some previously introduced IL models based on weight regularization are proposed by Chaudhary et al. [18], Aljundi et al. [5], and Castro et al. [22].

Distillation Strategies Distillation is a regularisation approach focused on macro-protection. It limits the output value of both the new and old models. It allows information from the previous model to be incorporated into the current model, hence, CF is partially reduced.

Hinton et al. [23] suggested knowledge distillation (KD) to minimise knowledge loss. The KD equation may be stated as follows using the softmax output layer:

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_i/T)}$$

where q_i is the probability of the ith class and z is the preceding layer's logit. T is a temperature coefficient that is normally set to 1 during the inference stage.

Some of the studies that propose IL models based on distillation strategy are proposed by Wu et al. [7], Zhang et al. [24], Lee et al. [25].

One of the main advantages of regularization strategies is that they do not require extra storage during each incremental stage. However, most regularization strategies struggle when there are numerous incremental stages/tasks to learn (especially weight regularization strategies). The main reason for this poor performance is the large number of required regularization terms, which may prevent many weight parameters from updating, resulting in intransigence.

2.2.2 Architectural strategies

In architectural strategies, multiple classifiers are trained for every sequential incremental task. Then, during the inference stage, a selector decides which one is the model best suited for the task at hand.

Roy et al. [26] created a hierarchical framework with a tree structure. IL can be achieved by adaptively altering the tree's leaves. Although this strategy reduces catastrophic forgetting to some extent, it appears to take up more storage space; hence the model cannot be trained efficiently.

Other techniques, such as Progressive Neural Networks (PNN), rely on iteratively growing the network. But, most such methods cannot efficiently use the network's capacity. Mandivarapu et al. [27] propose a solution to this problem using the Self-Net model, which encodes a group of low-dimensional weights learned from multiple tasks using an auto-encoder.

Some previously introduced IL models using architectural strategies were proposed by Polikar et al. [28], Rusu et al. [29], Yoon et al. [30].

Thanks to their multiple connected classifiers, architectural strategies are beneficial for mitigating catastrophic forgetting while learning new knowledge. However, with each incremental iteration, their number of parameters expands, which may increase the model's training time for later iterations.

2.2.3 Variational continual learning

Variational continual learning (VCL) is built on the Bayesian inference framework. VCL was first proposed by Nguyen et al. [31], in an attempt to combine online Variational Inference (VI) and the latest advancements in Monte Carlo VI for neural networks.

Farguhar and Gal [32], propose Variational Generative Replay (VGR), a variational inference extension of Deep Generative Replay (DGR) based on the Bayesian online learning paradigm, to complement variational continual learning.

Some other interesting IL models using variational continual learning are proposed by Chen et al. [33], Adel et al. [34], and Ebrahimi et al. [35].

VCL is a natural way of capturing past information learned by a neural network in the form of a prior, which mitigates catastrophic forgetting. Its main drawback is that the estimation of prior that effectively captures past information is a big challenge. Furthermore, VCL is computationally expensive when the number of variables involved in Bayesian inference is large.

2.2.4 Rehearsal approaches

Rehearsal and pseudo-rehearsal approaches use retrospective knowledge to mitigate the effects of catastrophic forgetting (CF). The rehearsal strategy enables the IL model to re-use the previous knowledge while acquiring new knowledge by keeping in memory a subset of the data learned in the past. In pseudo-rehearsal techniques, upon learning new knowledge, the model creates some pseudo data that closely matches the data distribution of old data. Then, during each incremental learning iteration, both the new data and the "old" data (or pseudo-data in the case of pseudo-rehearsal) is used to train the incremental model.

One of the earliest rehearsal-based deep incremental learning model *Incremental Classifier and Representation Learning (iCaRL)* was proposed by Rebuffi et al. [4] and employs a CNN for the incremental classification of images from the CIFAR-100 and ImageNet ILSVRC 2012 datasets.

Wu et al. [36] propose a rehearsal-based IL technique that uses knowledge distillation for the incremental classification of images from the CIFAR-100, Flower-102, and MS-Celeb-1 M-Base datasets. Their model uses vanilla generative adversarial networks to cater to the distribution difference in the exemplars kept in memory.

Zhang et al. [24] propose Deep Model Consolidation (DMC) algorithm based on pseudo-rehearsal incremental learning technique. The proposed approach trains two individual models (for old and new classes) and then combines them via a novel double distillation strategy. The combined model is further consolidated by exploiting publicly available unlabelled auxiliary data representative of both old and new classes.

Masarczyk and Tautkute [37] propose a pseudo-rehearsal incremental learning model that uses a two-step optimization process to generate synthetic data via meta-gradients, which, when learned in a sequence, does not result in catastrophic forgetting. In the first step, the model uses generative neural networks to create a synthetic sequence of tasks to evaluate the learner model in IL scenarios. The learner model, once trained on synthetic data, is evaluated on real data. The loss obtained on real data is used to fine-tune the parameters of the generative network. The process continues whenever a new task or a class is encountered. The proposed model is trained to incrementally classify images from the Split-MNIST dataset.

Some previously introduced rehearsal and pseudo-rehearsal incremental learning models were proposed in Shin et al. [38], Kemkar and Kanan. [39], and Hou et al. [40].

Rehearsal techniques work well and inherit a long history of success. They are considered efficient and effective to tackle CF because, by keeping only a limited amount of previously learned data, CF is reduced significantly. Though rehearsal-based incremental learning models are simple to implement, they require setting carefully several hyper-parameters to achieve optimal performance: the number of initial classes, memory size (for the instances from old classes), the batch size for new classes, etc. One of the goals of this paper is to give a few recommendations for setting these parameters in the context of incremental document classification.

2.3 Discussion

The performances of automatic document classification systems for companies have greatly improved in recent years [41, 42]. Nonetheless, the requirements concerning training

datasets, the time to set up, and the cost to keep up with the changes as the domain grows still pose a serious challenge for the large-scale practical application of these systems [43, 44]. Furthermore, the evolving nature of business document datasets makes it difficult for static machine learning models to achieve optimal performance in the long run. Incremental learning models are more suited for training machine learning models for business document classification.

Table 2 summarizes incremental learning approaches discussed in the previous section. To ease their comparison, we describe the advantages and/or drawbacks of each category.

Several researchers have undertaken the task of applying incremental learning techniques for document classification [46–50]. However, most of the existing research works report performance of incremental learning models for document classification in terms of a single performance metric such as accuracy or F1 measure on a static test set. Furthermore, to the best of our knowledge, none of the existing works investigate the performance of rehearsal-based incremental learning models for document classification, nor how to set its main parameters (memory size, best number of new classes to add at each iteration, the effect of the order in which these new classes appear, etc).

This research work proposes a rehearsal-based incremental learning protocol for text document classification using the weight-sharing strategy of transformer models layers across multiple incremental iterations. Our choice to use rehearsal-based approaches for document classification is motivated by (i) the lack of existing rehearsal-based incremental learning model for document classification, (ii) the simplicity and explainability of rehearsal-based models, as they only require mixing up instances from new and old classes for every training cycle, and (iii) the absence of evidence regarding optimal values for various rehearsal-based IL parameters, which settings greatly affects the overall performances of the model.

For this work, we chose not to consider pseudo-rehearsal approaches since they are dependent on the quality of the generative models for the generation of data for representing the old classes. Moreover, in the case of imbalanced datasets where some classes are not well represented, the quality of generated pseudo-data is subpar to real data used in rehearsal strategies. [51].

In the rest of this paper,

- We propose a rehearsal-based incremental learning model for document classification;
- Using different datasets, we compare extensively its performances with its static counterpart, and investigate the effects of class imbalance on both models;
- Experiments are performed to analyse the effect of various parameters for the incremental learning model on real-life training scenarios for incremental document classification. To this end we attempt to find the answers to the following questions for various training scenarios: (i) What is the effect on the overall performance, of the order in which new classes appear during training, (ii) what is the optimal batch size of previous classes and the number of instances from previous classes to keep in memory, and (iii) what is the optimal number of new classes to add during each incremental training cycle.

The next section presents the proposed rehearsal-based incremental learning protocol for document classification and the methodology adopted to evaluate the model performance. Recommendations are also provided about which values to use for the main hyper-parameters of such rehearsal-based approaches in the case of document classification.

3 Proposed approach and methodology

Most previously proposed approaches typically involve a traditional, static machine learning pipeline where models are trained once using the complete dataset and tested on a static test set. Unlike such approaches, this research work proposes rehearsal-based incremental learning protocol that involves model training in multiple iterations using a subset of data during each iteration.

Rehearsal-based incremental learning models are simple to implement. However, for the most optimal performance, the values of some of their hyper-parameters are required to be identified before training, which is one of the goals of this research work.

The rest of this section explains the datasets, the proposed protocol, and the methodology we adopted to evaluate its performances and select its best hyper-parameter values in the context of document classification. A similar methodology could be employed for other final applications.

3.1 Datasets

Two datasets are used to train and evaluate our incremental learning models. The first dataset is a real-world French language dataset consisting of grayscale document images from 47 classes.

In total, there are 23,577 images in the dataset, among which 15,491 documents are in the training dataset, 2203 documents are in the validation dataset, and 5883 in the test dataset. The orientation of the images in the dataset is mostly horizontal. It is important to note that the orientation is not an indicator of any particular class label.

This dataset is highly imbalanced, as some of its least represented classes have only a single instance in the training set, while the most represented classes have up to 2940 instances. Table 3 depicts the class distribution in training, test, and val-

Table 2 Various categories of incremental learning approaches along with their advantages and limitations

Method	Description	Advantages	Limitations	Existing works
Rehearsal approaches	A subset of data from previous stages is kept when training incrementally the model for new tasks. In the case of pseudo-rehearsal approaches, a subset of data from previous stages is not stored, instead, it is generated on the fly for new tasks	Simplicity, high explainability and a long history of success. Only requires a subset of previous data while learning the new knowledge	Memory consumption to keep the subset of previous data and longer training time are a few challenges that often arise with this category	[4, 24, 36–40]
Regularization approaches	Reduces catastrophic forgetting by using a unique regularization term that restricts the modification of parameters beyond an acceptable point	The regularization models are more stable compared to other approaches as the regularization term bounds the amplitude/direction of parameter changes	The regularization approaches suffer the consequences of limited growth (slow update of weight parameters), and possibly, intransigence	[5, 7, 18, 20–25]
Architectural strategy	Typically, multiple classifiers are trained on different subsets of data. The best-suited classifier from the learned model is then chosen at the inception phase	A dedicated classifier for each learning stage is more suited for eliminating inter-task confusion and mitigating catastrophic forgetting	This technique requires continuous model expansion and continuous parameter change and might suffer from large training time for later iterations	[26–30]
Variational continual learning (VCL)	VCL employs Bayesian inference to integrate the knowledge about model parameters collected from existing data (prior) with the knowledge from present data (the likelihood)	VCL is a generic approach that works well with both deep descriptive and deep generative models in complex real-time environments	VCL is typically based on Bayesian inference, which is often hard to estimate accurately, and so approximations are often needed. VLC can also be computationally expensive	[31–35, 45]

Table 3 Class distribution for the imbalanced private dataset

Class	Training set	Test set	Validation set
Account 1	2940	1135	457
Bank 1	2540	970	365
Business 1	1640	649	304
Bank 2	940	392	239
Business 2	940	376	188
Bank 3	800	288	64
Account 2	680	244	54
Account 3	580	208	45
Account 4	570	204	43
Account 5	420	152	39
Business 3	400	149	48
Business 4	360	133	39
Business 5	330	122	36
Account 6	250	91	23
Bank 4	230	83	22
Account 8	200	75	25
Account 7	200	76	29
Bank 5	189	70	21
Legal 1	160	57	13
Account 9	130	49	19
Account 10	120	46	19
CGV	120	44	13
Business 6	110	43	18
Account 11	85	30	7
Account 12	80	28	7
Account 13	74	26	7
Insurance 1	71	26	7
Account 14	67	24	7
Mail 1	52	19	7
Business 7	40	15	7
Amount document	34	12	4
Legal 2	24	8	3
Mail 2	24	9	3
Legal 3	22	8	3
Verso	19	7	5
Legal 4	17	6	2
Account 15	9	3	2
Insurance 2	6	2	2
Bank 6	5	2	1
User Info	4	1	1
Mail 4	2	1	1
Mail 3	2	0	1
Bank 7	1	0	1
Mail 5	1	0	1
Mail 7	1	0	0
Legal 5	1	0	0
Mail 6	1	0	1

Class names anonymised for privacy concerns

idation sets. It has to be noted that, for 6 out of 47 classes (the ones with the fewest samples), there is no document in the test dataset, but there is one document each for 4 of those classes in the validation set. Despite the absence of samples in the test dataset for those 6 classes, we chose to keep all 47 classes in the dataset because (i) we chose to adopt a realistic scenario where some classes that were learned might never occur for a certain customer and (ii) it is interesting for real-life scenarios to observe the effect of such "never-seen" classes on the overall performances of the model.

Though our private dataset consists of real-world documents, it cannot be accessed publicly. Hence, we also performed experiments on the publicly available RVL-CDIP dataset which consists of 400,000 grayscale document images belonging to 16 classes, with 25,000 images per class [52]. Different from our private dataset, RVL-CDIP is thus a balanced dataset, and it is in English. It contains three subsets: training, testing, and validation. The training set contains 320,000 images, while the validation and test sets contain 40,000 images each.

3.2 Proposed approach

Figures 1 and 2 respectively depict the static and rehearsal-based incremental learning model for document classification proposed in this research work.

As shown in Fig. 1, the static document classification model is a traditional machine learning model where the input to the model is the text extracted from document images from all the classes in the dataset via an Optical Character Recognition (OCR) tool; here we chose ABBYY fine reader for its excellent performances in practice.[1]

The vector representation for text documents in the RVL-CDIP dataset is generated via the DistilBERT [53] model which is a fast, smaller, and lighter version of the BERT model [54]. For our french language private dataset, we switched to the transformer-based Flaubert language model [55]. The Flaubert model is pretrained using a french language corpus.

We chose to use DistilBERT and Flaubert transformers for text representation, because our preliminary experiments showed that they yield excellent performance for the static

[1] ABBYY fine reader (https://www.abbyy.com/ocr-sdk/features/ocr/).

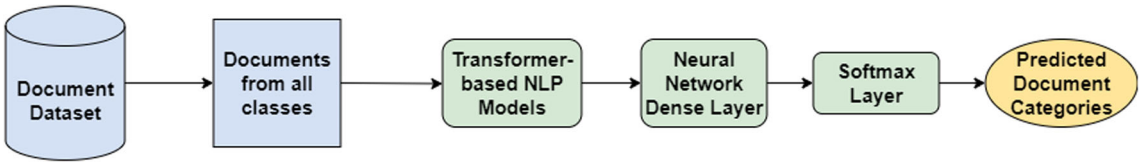

Fig. 1 Proposed approach for document classification using static deep learning model

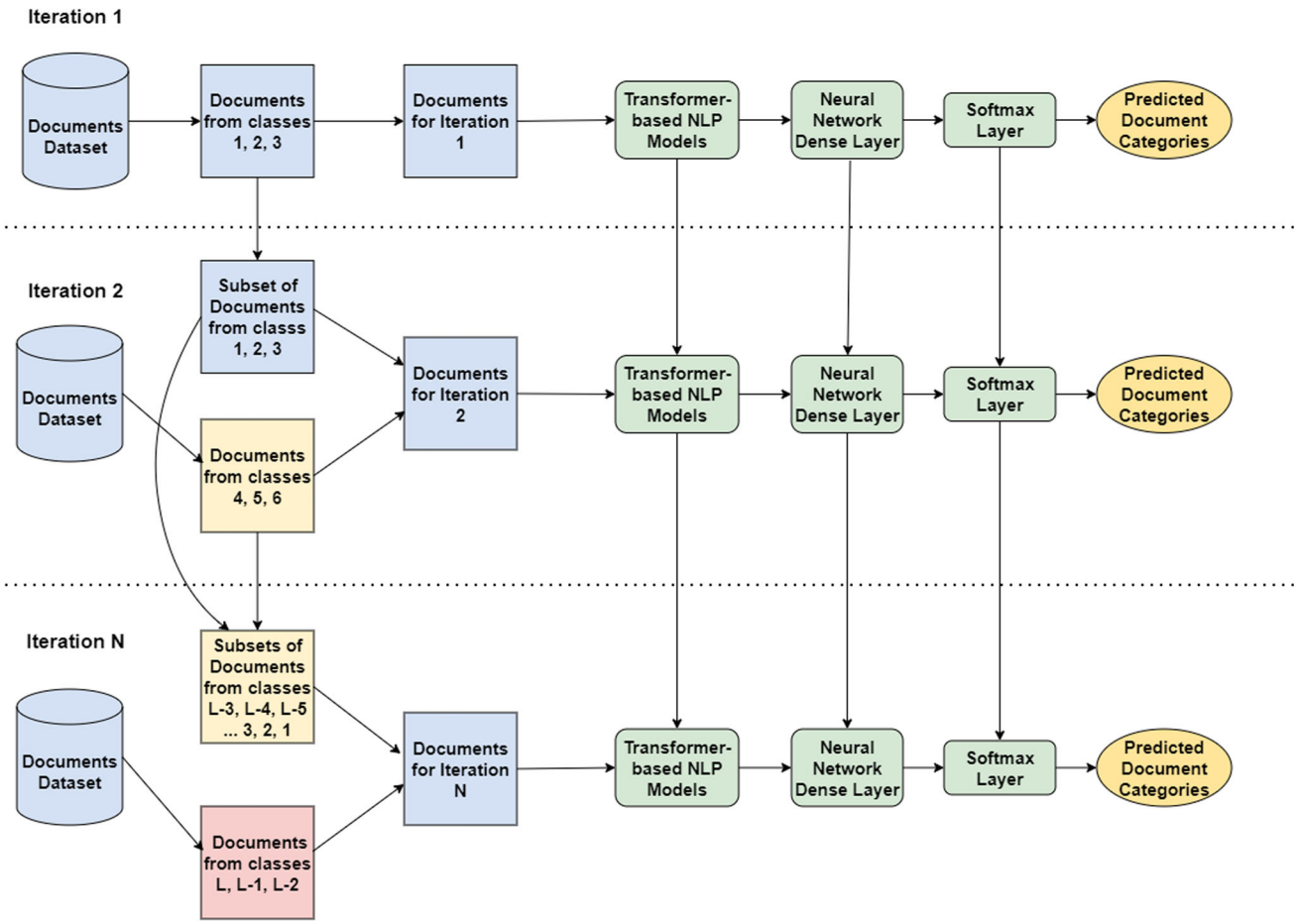

Fig. 2 Proposed approach for rehearsal-based incremental document classification. In this diagram, the number of initial classes (BC) and new classes per iteration (NC) is arbitrarily set to 3. L represents the total number of classes. N represents the total number of training iterations

classification of documents written in English and French languages, respectively, using deep learning models.

The vector representations for input documents, as returned by the DistilBert and Flaubert transformer models, are passed to a dense layer of a fully connected neural network for further fine-tuning. Since we have a multi-class classification problem where the model output is a label for one of the several document categories, a Softmax layer is added to make final predictions. The categorical-cross entropy loss function is used to calculate loss and the Adam optimizer is used to optimize the model.

Our rehearsal-based incremental learning model, as depicted in Fig. 2, is trained in multiple iterations where the workflow of each individual iteration is very similar to the static model. However, unlike the static model, for each iteration, document images from only a small number of new classes (3 in the illustrative figure) are used for training, along with the subset of document images from classes used for training the model in the previous iterations.

For example, in Fig. 2, in the first iteration, all the documents from classes 1, 2, and 3 in the training dataset are used for training. In the second iteration, the model is trained using documents from classes 4, 5, and 6, along with a subset of documents from classes 1, 2, 3. This process continues until the model is trained using documents from L classes across N iterations, where L is the total number of classes and N is

Table 4 Simulation strategy for the proposed approach for rehearsal based incremental classification of documents

IEC	Exp1		Exp2		Exp3		Exp4		Exp5	
	BC	NC	BC	NC	BC	NC	BC	NC	BC	NC
50	2	1	2	2	3	3	4	4	5	5
100	2	1	2	2	3	3	4	4	5	5
150	2	1	2	2	3	3	4	4	5	5
200	2	1	2	2	3	3	4	4	5	5
250	2	1	2	2	3	3	4	4	5	5

the total number of training iterations. For the last iteration N, the model is trained using the documents from L, L-1, and L-2 classes, along with a subset of documents from all the previous classes L-3, L-4, L-5 up to classes 3, 2, and 1.

Another important difference is that, in the rehearsal-based incremental learning model, the weights learned during previous iterations in the transformer-based NLP models, the dense neural network layers, and the softmax layers are passed to the subsequent iterations. This approach allows retaining the knowledge learned during previous iterations to classify documents from the previously encountered classes. By preserving the weights learned in transformer-based NLP models, the dense neural network layers, and the softmax layers across iterations, our rehearsal-based incremental learning model achieves a seamless integration of prior knowledge, enabling effective classification of documents from familiar classes. As shown in Sect. 4.1, this continuity in learning (as opposed to static models) contributes significantly to the model's performance and ensures the retention of valuable insights from past iterations.

3.3 Experimental protocol

One of the goals of this research work is to study the effect of the batch size of new classes used for each iteration of incremental learning model. In this regard, a total of 5 experiments are performed as summarized in Table 4. The number of base classes (BC) and the number of new classes (NC) have been chosen arbitrarily for each experiment.

In the rest of this article, the base classes (BC) refer to the classes that are used to train the incremental learning model for the first time, and the new classes (NC) refer to the classes that are used to train the model for the next iterations, and instances from existing classes (IEC) will refer to the number of instances from each of the classes that have been used to train the incremental learning model in previous iterations. For each experiment, 50, 100, 150, 200, and 250 are considered possible values for the number of instances from existing classes (IEC). In the cases where an existing class contains fewer instances than the IEC value, all the instances from this class are kept in memory.

The number of iterations (NoI) for incremental training is calculated as follows:

$$NoI = Ceil\left(\frac{TotalClasses - BC}{NC} + 1\right) \tag{1}$$

For incremental training in real scenarios, new classes can have more, less, or a similar number of documents as the classes that were already used in previous training iterations. Therefore, to study the effect of the order of addition of new classes on the overall performance of the incremental learning model, three training strategies are adopted:

- *Most Frequently Occurring Classes First* The base classes (BC) and new classes (NC) for each incremental iteration are selected in the descending order of the number of instances per class.
- *Least Frequently Occurring Classes First* The base classes (BC) and new classes (NC) for each incremental iteration are selected in the ascending order of the number of instances per class.
- *Random Addition of New Classes* Document classes are randomly selected for all the training iterations of the incremental learning model (without taking into account their occurring frequency).

The motivation behind selecting these three strategies is to see how the order of addition of new classes affects the overall model performance, and what are the most optimal values for the parameters number of base classes (BC), new classes (NC), and instances from existing classes (IEC), which return the best performance for the aforementioned scenarios. These training strategies can also help us understand the effect of class imbalance, which is often present in real-life document datasets, including our highly imbalanced private dataset.

3.4 Evaluation strategies and measures

Two different but related evaluations strategies are adopted for (i) the experiments that compare the rehearsal-based incremental learning model with the static deep learning model for document classification, and (ii) for the experiments that study the effect of class order, and of the hyper-parameters BC, NC and IEC.

3.4.1 Evaluation strategy for comparison with the static learning model

To obtain baseline results for our static deep learning model, the dataset is divided into training, validation, and test sets. Static deep learning models are trained using the complete training set in a single pass. The best model is selected via the model performance on the validation set which is also

obtained *via* a single pass prediction. Finally, for the sake of comparison, predictions are made on the independent test set.

To compare rehearsal-based incremental learning models with the static deep learning models, depending upon the number of iterations (NoI), the training and validation sets are divided into multiple sub-training and sub-validation sets. For instance, if the number of incremental iterations is 10 as in the case of Exp5 on our private dataset ($TotalClasses = 47$, BC=5, NC=5), the number of sub-training and sub-validation sets is 10. The incremental learning model is trained and validated in 10 iterations using these sub-training and sub-validation sets. The final model performance for both incremental and static models is compared by using the same fixed test set.

We chose accuracy as a performance metric since it has been widely used in the literature for the performance evaluation of document classification systems. Furthermore, since our private dataset is highly imbalanced, we chose to use, on top of accuracy, the F1 measure since it provides better insights about the model performance in case of imbalanced datasets.

3.4.2 Evaluation strategy to study the effect of class order, BC, NC and IEC values

To study the effect of batch sizes of new classes (NC) and base classes (BC), the order in which classes are used for training, and the number of instances from existing classes (IEC), on the incremental learning model performance, a static test set is not required. Hence, the test and validation sets are merged, resulting in updated validation sets. Depending upon the number of iterations (NoI), the training and validation sets are further divided into sub-training sets and sub-validation sets. The model performance is evaluated using the average of accuracies and F1 measures for all the sub-validation sets.

The code for the proposed approach and experiments is available online.[2]

4 Results and discussion

This section contains the results and related discussion for two kinds of experiments:

- the experiments performed for comparing the incremental learning model with the static deep learning model, and for studying the effect of class imbalance on both models;
- the experiments to study the effect of class order and of the values of hyper-parameters BC, NC, and IEC on the incremental learning model.

In order to evaluate our proposed methodology against the current state-of-the-art, we have chosen the study conducted by Voerman et al. [50] as the baseline approach. Their approach addresses the challenge of low-represented class classification by employing a cascaded system. This strategy leverages deep learning networks for major classes, ensuring high precision while employing specialized architectures such as few shot-learning for rare classes. Their approach works in two stages: (i) documents are classified using deep neural networks, and only predictions with high confidence are selected; and (ii) for predictions with low confidence, a specialized architecture such as few-shot learning is used. The final prediction is based on the global confidence of all the systems in the cascade. A major drawback of their approach is that it requires training multiple neural networks in a cascade system. Predictions are generated in series based on the outputs of these multiple neural networks, leading to a potential slowdown in the model's performance.

This selection of the approach from Voerman et al. [50] as a baseline is based on two primary reasons: firstly, their research represents the sole existing work focusing on incremental classification of business document streams, and secondly, it utilizes the same datasets employed in our study. The results for experiments involving the random addition of new classes are compared with the baseline model since these experiments resemble the approach adopted in the baseline model.

In the next section, we will compare the performance of our incrementally trained model with that of their static counterpart.

4.1 Comparison with static model

This section contains results of the experiments where models are trained incrementally and evaluated on the same fixed test set that is used for the evaluation of statically trained models, i.e. the test set of the dataset used (RVL-CDIP or our private dataset), as detailed in Sect. 3.1).

4.1.1 Results for private dataset

Tables 5, 6 and 7 depicts the results for the experiments where models are incrementally trained via the private dataset when adding the most frequent classes first, least frequent classes first, and using random class addition approach, respectively. These results show that in the scenario where the model is incrementally trained using the most frequent classes first or the random-class-addition, a maximum accuracy of 97.65% is achieved, which is only slightly less than the accuracy

[2] Link to the code and appendix: http://bit.ly/3hC6ved.

Table 5 Percentage accuracies for private dataset on test set when incremental learning is performed using most frequently occurring classes first

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	96.20 (0.95)	97.16 (0.96)	97.11 (0.95)	97.22 (0.96)	97.34 (0.96)
Exp2	95.88 (0.94)	96.97 (0.94)	97.38 (0.95)	96.20 (0.95)	97.56 (0.96)
Exp3	95.98 (0.93)	96.97 (0.94)	97.21 (0.96)	**97.65 (0.97)**	97.00 (0.96)
Exp4	95.69 (0.94)	96.61 (0.95)	97.29 (0.96)	97.07 (0.96)	97.53 (0.96)
Exp5	95.83 (0.94)	97.31 (0.96)	97.31 (0.96)	97.26 (0.96)	95.81 (0.93)

Weighted F1 in brackets. Bold indicates highest values. Recalls: (i) for the values of the parameters BC and NC inside each experiment, please refer to Table 4 and (ii) accuracy with Static model = 98.23%

Table 6 Percentage accuracies for private dataset on test set when incremental learning is performed using least frequently occurring classes first

Exp.Id	IEC Values				
	50	100	150	200	250
Exp1	96.99 (0.95)	96.34 (0.95)	97.19 (0.95)	97.31 (0.96)	97.00 (0.95)
Exp2	96.48 (0.95)	96.15 (0.95)	97.41 (0.96)	97.24 (0.95)	96.70 (0.95)
Exp3	95.34 (0.94)	95.24 (0.95)	96.34 (0.95)	**97.46 (0.95)**	97.28 (0.96)
Exp4	94.28 (0.94)	96.10 (0.95)	96.37 (0.96)	96.51 (0.95)	97.39 (0.96)
Exp5	94.86 (0.93)	96.80 (0.95)	96.60 (0.95)	97.40 (0.96)	96.29 (0.94)

Weighted F1 in brackets. Bold indicates highest values. Recalls: (i) for the values of the parameters BC and NC inside each experiment, please refer to Table 4 and (ii) accuracy with Static Model = 98.23%

Table 7 Percentage accuracies for private dataset on test set when incremental learning is performed via random addition of new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	96.46 (0.95)	96.99 (0.95)	97.02 (0.96)	96.87 (0.96)	97.22 (0.96)
Exp2	95.00 (0.93)	96.36 (0.94)	96.88 (0.96)	**97.65 (0.97)**	97.45 (0.95)
Exp3	96.56 (0.95)	96.12 (0.96)	96.90 (0.96)	97.14 (0.95)	97.04 (0.96)
Exp4	97.09 (0.96)	95.47 (0.95)	97.11 (0.96)	96.88 (0.95)	97.36 (0.96)
Exp5	96.54 (0.95)	95.39 (0.95)	96.78 (0.96)	97.48 (0.96)	96.56 (0.95)

Weighted F1 in brackets. Bold indicates highest values. Recalls: (i) for the values of the parameters BC and NC inside each experiment, please refer to Table 4 and (ii) accuracy with Static Model = 98.23%

achieved by the static model (98.23%). For the least-frequent-classes-first scenario, an accuracy of 97.46% is achieved.

Thus, in the rest of this section, we mainly consider the incremental scenario where new classes are added randomly (independently of their occurrence frequency) because it gives the best results, and it is certainly the most realistic scenario for most real-life applications where the user does not necessarily know the number of instances for each class.

Selecting the best rehearsal-based IL model parameters for comparing with static model

Let us now focus on selecting the best rehearsal-based IL model hyperparameters, to compare with its statically trained counterpart. For this sake, let us start by studying the effect of the parameter IEC: the number of instances from each existing class to be kept in memory when classes are added in random order. Analysis of variance test (ANOVA) [56] tests were carried out to compare the accuracies obtained via various experiments and IEC values for IL models trained with random addition of new classes (accuracies shown in Table 7). The test showed that the difference among the results obtained for all the experiments using various IEC values is statistically significant, with $p - value < 0.05$ ($p - value = 0.0007$), meaning that the value of IEC has an impact on the model's performances.

To further explore the effect of IEC values, T-Tests were performed to determine the significance of the best-case result obtained via an IEC value of 200. To this end, pair-wise T-tests were performed to compare the accuracies obtained with IEC = 200 on one hand, and the results obtained via the remaining IEC values on the other hand. The results show that the difference between the accuracies obtained using IEC values of 50 and 100 on one hand, and 200 on the other hand, are statistically significant. However, the difference in the accuracies obtained with IEC values of 150, 200, and 250 is not significant. This shows that IEC values can be divided into two groups based on statistical significance: (50, 100) and (150, 200, 250), with the latter reaching the best performances.

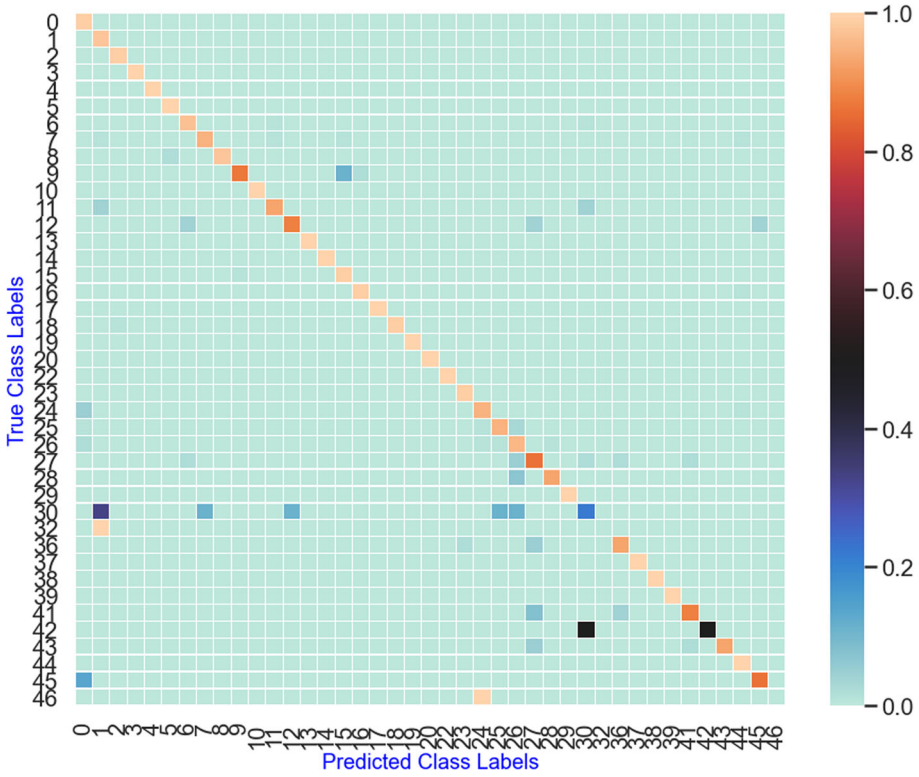

Fig. 3 Confusion matrix for static model for private dataset

Now that we selected the optimal values of IEC, let us focus on the effect of the parameters BC and NC: the numbers of base classes and new classes at each iteration respectively, when classes are added in random order. ANOVA tests were performed to compare the accuracies obtained for various experiments (Exp1, Exp2, Exp3, Exp4, and Exp5, with different values of BC and NC as detailed in Table 4), for prediction on a static test set. This test showed that the differences between these accuracies are not statistically significant at $p < 0.05$ ($p = 0.96$). Thus, the numbers of base classes (BC) and new classes (NC) do not significantly impact the model's performance. Similar results were obtained for IL models trained using most-frequent and least-frequent classes first.

Comparing rehearsal-based IL model with the static model, in the presence of class imbalance

To further compare the behaviour of statically and incrementally trained models using a static test set, we perform error analysis on the results obtained via the static model and the incrementally trained model that returns the highest accuracy with the random addition of new classes (Exp 2, BC = 2, NC = 2, IEC = 200). Figures 3 and 4 depict confusion matrices for all the classes in the static test set, predicted via static and incrementally trained models, respectively. To help analyze the confusion matrices, the mappings between the class IDs in the confusion matrix and the names of the document categories in the corresponding datasets are depicted in Table 8

These confusion matrices show that the classification behaviours of incremental and static learning approaches are quite similar, even though, on average, the incremental learning model misclassifies a given class with fewer other classes than its statically trained counterpart. Indeed, for the static model, on average the number of labels assigned to one class is 3.04 (including the true label), whereas this number is 2.31 for the incrementally trained model. The reason could be that a statically trained model is trained using all classes at once, hence leading to confusion with more other classes, for one given class. On the other hand, with rehearsal-based incrementally trained models, new classes are learned in multiple iterations with fewer classes per iteration, particularly in initial iterations, which might lead to confusion with fewer other classes, for one given class.

This assumption is further supported by the observation that in the case of an incrementally learned model, misclassification mostly occurs among the classes that are trained together for the first time in the same training iteration. For instance, when, for incremental training, classes 42 (Insurance Daily Allowance) and 1 (Account Dues) were trained as new classes in the same training iteration, class 42 was misclassified as class 1 for 50% of instances (see Fig. 4) whereas, with the statically trained model, class 42 is mostly confused with class 30 (see Fig. 3).

Furthermore, comparisons of prediction accuracies between individual classes for statically trained and incrementally trained models show that prediction accuracies for both static and incremental models are similar in most cases, with a very slight difference in some cases. The most notable

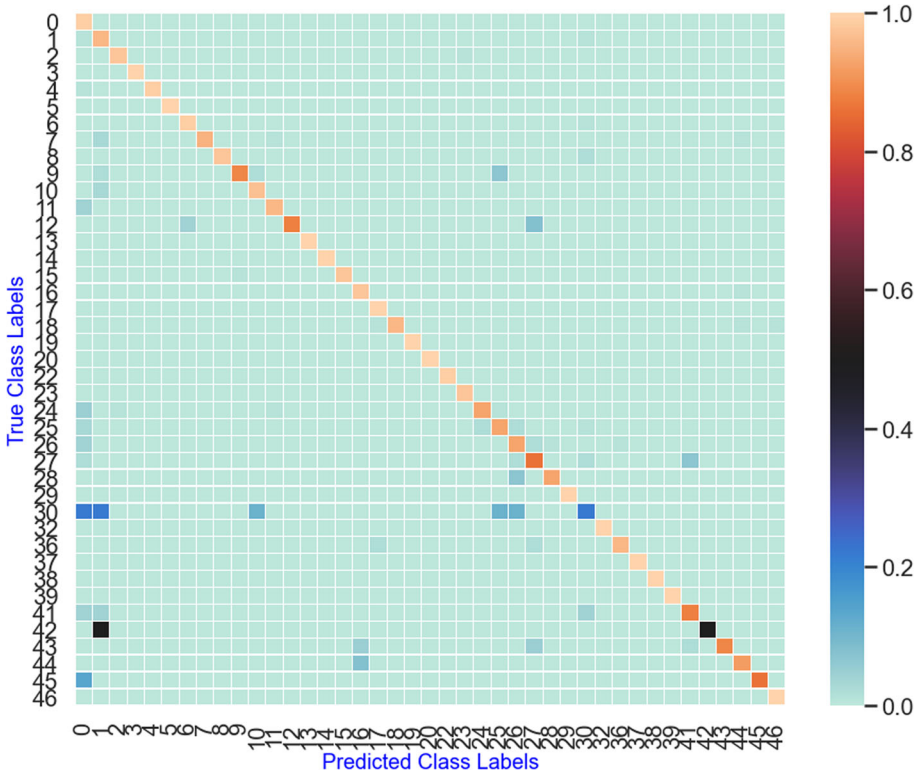

Fig. 4 Confusion matrix for incrementally learned model on private dataset (Exp 2, BC = 2, NC = 2, IEC = 200)

differences between accuracies is observed for classes with Ids 32 (Mail Account Dues) and 46 (ID i.e. identity documents), for which the incremental learning model returns an accuracy of 100%, whereas the static model fails to correctly predict any instance for these classes. These results are striking, given that we use the same test set for static and incremental learning (as explained before).

Further investigation reveals that classes with Ids 32 and 46 are among the least represented both in the training and test sets (due to class imbalance in our private dataset). The reason for the better performance of incremental models for least represented classes could be that in the case of the incremental learning model, the class imbalance in each iteration is less compared to static models trained on the whole dataset (in particular, thanks to the use of a bounded-sized memory).

4.1.2 Results for RVL-CDIP dataset

For the RVL-CDIP dataset, the results for the incrementally trained model are presented in Table 9. The results show that a maximum performance of 78.50% is achieved when the IEC (instances from existence classes) value is 200 and batch size for base classes (BC) and new classes (NC) is 5 (Exp5). The performance achieved via the incremental model is slightly better than the static model performance on the RVL-CDIP test set: 76.49%.

Selecting the best rehearsal-based IL model parameters for comparing with static model

Just like for our private dataset, ANOVA tests show that the parameter IEC is statistically linked to the accuracy (here $p - value = 0.00009$), that the effect of the number of base classes (BC) and new classes (NC) is not statistically significant at $p - value = 0.31$, and pair-wise T-tests show that based on statistical significance, IEC values can be divided into two groups: (50, 100) and (150, 200, 250), the latter reaching the best performances.

Thus, in the rest of this section, we compare the static model with the rehearsal-based IL model that gives us the best accuracy on the RVL-CDIP static test dataset when classes are added in random order: IEC=200, BC=NC=5.

Comparing rehearsal-based IL model with the static model (balanced classes)

Figures 5 and 6 give the confusion matrices for all the classes in the RVL-CDIP test set predicted via static and incrementally trained model, respectively. The comparison of these two confusion matrices reveals that, on average, the static model confuses a given class with more other classes (10.32) compared to the incremental learning model (8.06). This behaviour is similar to the results obtained on our private dataset.

The comparison of the individual accuracies depicts that overall, the performance of the incremental learning model

Table 8 Mappings between class IDs and document categories in the private (French) and RVL-CDIP(English) datasets

Id	Private dataset	RVL-CDIP dataset
0	Account 1	Letter
1	Account 2	Form
2	Account 3	Email
3	Account 4	Handwritten
4	Account 5	Advertisement
5	Account 6	Scientific report
6	Account 7	Scientific publication
7	Account 8	Specification
8	Account 9	File folder
9	Account 10	News article
10	Account 11	Budget
11	Account 12	Invoice
12	Account 13	Presentation
13	Account 14	Questionnaire
14	Account 15	Resume
15	Bank 1	Memo
16	Bank 2	–
17	Bank 3	–
18	Bank 4	–
19	Bank 5	–
20	Bank 6	–
21	Bank 7	–
22	Business 1	–
23	Business 2	–
24	Business 3	–
25	Business 4	–
26	Business 5	–
27	Business 6	–
28	Business 7	–
29	Mail 1	–
30	Mail 2	–
31	Mail 3	–
32	Mail 4	–
33	Mail 5	–
34	Mail 6	–
35	Mail 7	–
36	Legal 1	–
37	Legal 2	–
38	Legal 3	–
39	Legal 4	–
40	Legal 5	–
41	Insurance 1	–
42	Insurance 2	–
43	CGV	–
44	Amount document	–
45	Verso	–
46	User profile	–

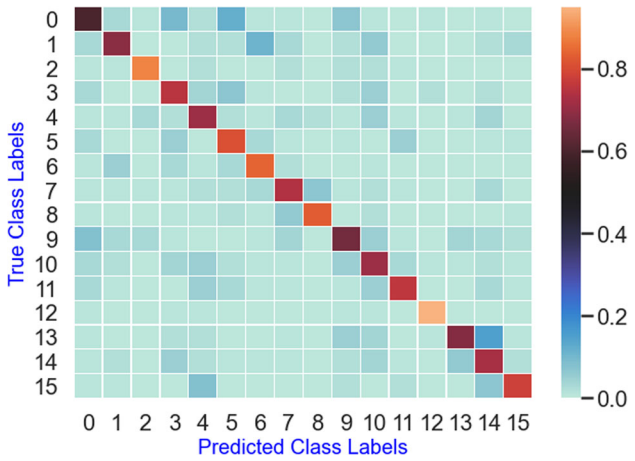

Fig. 5 Confusion matrix for RVL-CDIP dataset using static model. (Cells contain accuracy)

is slightly better than the static model on the same fixed test set. The reason could be that for balanced datasets such as the RVL-CDIP, in the case of incremental learning, the relationship between the feature and label set is learned several times (during multiple iterations), particularly for the classes in initial iterations.

Table 9 Percentage accuracies for RVL-CDIP dataset on test set when incremental learning is performed via random addition of new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	69.50 (0.68)	75.62 (0.74)	73.18 (0.73)	74.93 (0.72)	77.62 (0.76)
Exp2	67.06 (0.65)	74.37 (0.73)	73.62 (0.73)	76.37 (0.75)	78.18 (0.77)
Exp3	72.31 (0.71)	76.68 (0.75)	77.12 (0.76)	77.50 (0.77)	77.31 (0.76)
Exp4	61.43 (0.60)	73.50 (0.72)	72.43 (0.71)	72.68 (0.71)	74.50 (0.73)
Exp5	69.25 (0.65)	73.56 (0.71)	75.43 (0.74)	**78.50 (0.76)**	76.37 (0.75)

Weighted F1 in brackets. Bold indicates highest values. (Accuracy with Static model = 76.49%)

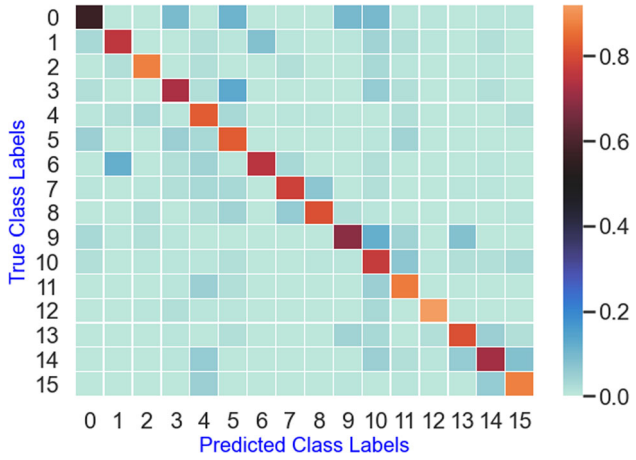

Fig. 6 Confusion matrix for RVL-CDIP dataset using incrementally learned model (Exp 5, IEC = 200)

4.2 Effects of class order, BC, NC, and IEC values for private dataset

This section presents results for the experiments performed on our private dataset (with class imbalance) to study the effects on the performances of incrementally trained models of (i) the order in which new classes are added, (ii) the batch size of the base and new classes (BC/NC) and (iii) the IEC (instances from existing classes).

In the previous section, we studied the effects of these hyper-parameters only with the aim of selecting the optimal model settings before comparing the incremental model to its statically trained counterpart (on a fixed test dataset). In this section, on the other hand, we are going to study more thoroughly the effects of these hyper-parameters on the IL model's performances at each incremental iteration, depending on the type of class considered (newly added classes *vs.* classes added in earlier iterations).

To this end, the model performance is evaluated against the average of accuracies and weighted F1 values for: (i) all the classes in validation sets for each iteration, (ii)"old" classes only, in validation sets for each iteration, and (iii) "new" classes only, in the validation sets for each iteration.

To study the effect of the order in which the new classes are added in each iteration, we compare three scenarios on our highly imbalanced private dataset: most-frequently-occurring classes first, least-frequent-occurring classes first, and random addition of new classes.

4.2.1 Private dataset–ost-frequently-occurring classes first

Table 10 depicts the average accuracies achieved on the validation sets for all the classes of the IL models trained using classes with the most number of instances first. As explained earlier, we consider NoI validation sets in total (one for each iteration), where NoI depends on the total number of classes and values of BC and NC, as detailed in Eq. (1). The results show that the overall best-observed performance (93.09%) is achieved for Exp1 (BC =2, NC=1) with an IEC value of 200.

Analysis of variance (ANOVA) test shows that the difference among the average accuracies obtained for each experiment using various IEC values is significant at $p < 0.05$ ($p = 0.002$). On the other hand, ANOVA tests show that the difference in average accuracies for various experiments ("Exp1"-"Exp5", see Table 4) is not statistically significant ($p - value = 0.95$), showing no significant effect of the number of base classes (BC) and new classes (NC) on the accuracy. Just like when used on a static test dataset, pairwise T-tests show that based on statistical significance, IEC values can be divided into two groups: (50, 100) and (150, 200, 250), the latter reaching the best performances.

Table 11 shows average accuracies achieved on the NoI validation sets for old classes only in the corresponding learning iterations (most-frequently-occuring classes first scenario). Just like when considering all classes in the validation set (see Table 10), the results obtained on "old classes" only depict that the overall best performance (93.13%) is achieved for Exp1 where the batch size for base classes (BC) is 2 and NC (new classes) is 1 with an IEC value of 200 (even though, similar to the average accuracies on all classes, ANOVA and pairwise T-tests show that there is no statistical difference between the average accuracies of Exp1-Exp5 and between IEC=200 and IEC values in the list {150, 200, 250}).

Average accuracies on overall NoI validation sets for newly added classes for IL models trained via the most-frequently-occurring classes first is depicted in Table 12. The highest average accuracy of 78.43% is achieved when the batch size of base classes (BC) and newly added classes (NC) is 5, with an IEC value of 150. In this specific case though, ANOVA tests reveal that there is no statistically significant effect of parameters IEC nor BC, NC on the average accuracy ($p - value = 0.072$ and $p - value = 0.58$ respectively).

The main reason for this lack of statistical significance could be the highly imbalanced nature of data at each iteration. This is because, during each iteration, the class imbalance remains very high, and the instances belonging to new classes remain much fewer than old class instances, irrespective of the IEC, BC, and NC values. Hence overall, there is no significant difference in the results obtained via different experiments ("Exp1"–"Exp5") and IEC values.

It is further observed that for all the experiments and all IEC values, the average accuracy values for old classes are higher compared to new classes. As an example, Fig. 7 demonstrates the accuracy values for the 10 iterations in Exp5 ($TotalClasses = 47$, BC=5, NC=5, see equation (1)) with IEC value of 150. The figure shows accuracies for old classes,

Table 10 Average of percentage accuracies on private dataset for all batches of validation sets when incremental learning is performed using most frequently occurring classes first

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	90.08 (0.89)	91.58 (0.89)	92.30 (0.91)	**93.09 (0.91)**	92.50 (0.91)
Exp2	89.27 (0.81)	92.85 (0.91)	91.56 (0.90)	91.41 (0.90)	92.03 (0.89)
Exp3	88.38 (0.87)	91.69 (0.89)	92.58 (0.91)	92.85 (0.90)	92.19 (0.91)
Exp4	89.78 (0.87)	89.95 (0.87)	92.85 (0.90)	92.34 (0.90)	92.31 (0.91)
Exp5	89.11 (0.87)	92.08 (0.90)	92.08 (0.90)	92.02 (0.89)	91.90 (0.89)

Weighted F1 in brackets. Bold indicates highest values

Table 11 Average of percentage accuracies for private dataset for old classes in all batches of validation sets when incremental learning is performed using most frequently occurring classes first

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	90.09 (0.90)	91.60 (0.90)	92.34 (0.91)	**93.13 (0.91)**	92.53 (0.90)
Exp2	89.31 (0.88)	92.91 (0.90)	91.63 (0.89)	91.45 (0.88)	92.12 (0.90)
Exp3	88.35 (0.87)	91.75 (0.89)	92.70 (0.90)	93.00 (0.91)	92.23 (0.90)
Exp4	89.86 (0.88)	90.02 (0.88)	93.10 (0.92)	92.84 (0.90)	92.54 (0.91)
Exp5	89.27 (0.87)	92.30 (0.90)	92.30 (0.90)	92.37 (0.90)	92.12 (0.90)

Weighted F1 in brackets. Bold indicates highest values

Table 12 Average of percentage accuracies for new classes in all batches of validation sets when incremental learning is performed using most frequently occurring classes first

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	69.29 (0.67)	72.28 (0.70)	78.03 (0.75)	74.27 (0.74)	71.35 (0.69)
Exp2	67.17 (0.65)	66.45 (0.65)	73.60 (0.71)	68.38 (0.67)	74.24 (0.73)
Exp3	60.26 (0.60)	68.31 (0.68)	70.95 (0.70)	74.25 (0.73)	72.16 (0.72)
Exp4	66.09 (0.65)	65.66 (0.64)	65.73 (0.65)	69.47 (0.68)	71.24 (0.70)
Exp5	56.71 (0.55)	78.43 (0.77)	**78.43 (0.76)**	62.33 (0.60)	64.87 (0.63)

Weighted F1 in brackets. Bold indicates highest values

new classes, and all classes during each iteration along with the rolling average values up to a particular iteration. The Figure shows that on average, the accuracy of classification for old classes is higher compared to new classes. The reason for such behaviour can be the fact that since models are trained via most-frequently-occurring classes in the initial iterations, the model has more instances from the old classes and hence it learns to classify old classes more effectively than new classes which are less represented within a training batch.

Another important observation is that for initial learning iterations, the accuracy for both new and old classes is higher compared to the later learning iterations. The reason can be the class order, since the initial iterations contain classes with more instances, resulting in models having more information to learn as compared to later batches. Also, the number of classes in the early iterations is lower compared to later iterations, and hence the model has to classify fewer classes with more information per class in the early iterations, leading to better performance.

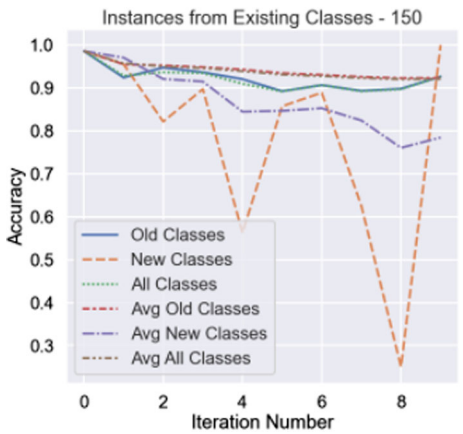

Fig. 7 Comparison of accuracies for old, new, and all classes for Exp5 with IEC = 150, for different iterations of incremental models trained on private dataset, using most-frequently-occurring classes first

Similar observations are made for different values of BC and NC, as shown in Appendix A (link to the appendix in the footnote.)[3]

[3] Link to the appendix: http://bit.ly/3hC6ved.

Table 13 Average of percentage accuracies for private dataset for all batches of validation sets when incremental learning is performed using least frequently occurring classes first

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	62.79 (0.60)	62.81 (0.60)	62.92 (0.61)	62.30 (0.60)	62.31 (0.60)
Exp2	60.43 (0.60)	60.99 (0.60)	62.25 (0.61)	61.52 (0.60)	61.45 (0.59)
Exp3	60.59 (0.59)	60.35 (0.59)	61.18 (0.59)	60.02 (0.58)	60.16 (0.58)
Exp4	64.98 (0.63)	63.73 (0.62)	63.10 (0.61)	64.49 (0.64)	64.74 (0.63)
Exp5	**67.02 (0.65)**	62.68 (0.61)	62.35 (0.61)	62.38 (0.60)	63.72 (0.62)

Weighted F1 in brackets. Bold indicates highest values

Table 14 Average of percentage accuracies for private dataset for old classes in all batches of validation sets when incremental learning is performed using least frequently occurring classes first

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	61.07 (0.60)	61.65 (0.59)	**61.71** (0.59)	60.64 (0.59)	60.63 (0.58)
Exp2	55.44 (0.54)	57.47 (0.56)	59.07 (0.58)	58.32 (0.57)	58.27 (0.56
Exp3	55.75 (0.54)	56.03 (0.54)	57.33 (0.56)	56.52 (0.55)	56.62 (0.55)
Exp4	56.18 (0.55)	56.76 (0.54)	56.50 (0.54)	57.58 (0.56)	58.09 (0.57)
Exp5	59.79 (0.58)	55.61 (0.54)	55.72 (0.54)	54.41 (0.53)	56.04 (0.54)

Weighted F1 in brackets. Bold indicates highest values

4.2.2 Private dataset–least-frequently-occurring classes first

Let us now focus on the scenario where the least frequently occurring size are learned first. Table 13 shows the average accuracies for all the validation sets for the IL models trained following this scenario. The results show that the best case accuracy of 67.02% is obtained for Exp5 (BC = 5, NC = 5) with an IEC value of 50.

In this scenario, hypothesis testing gives very different results compared to the most-frequently-occuring classes first scenario.

First, when using ANOVA tests to compare the average accuracies between experiments "Exp1"–"Exp5", the differences between different experiments are found to be statistically different ($p - value = 0.00004$). This behaviour can be explained by the fact that when training the model incrementally with the least frequently occurring classes first, increasing the batch size for new classes allows the model to train with more information. Also, with a higher number of classes in the case of the least-frequently-occurring training strategy, it is possible that even with few instances for each class, some classes are more easily recognizable, resulting in better overall average accuracy.

Second, ANOVA tests show that there is no significant difference between different IEC values. The reason could be that 18 out of the total 47 classes have less than 50 instances, hence for the earliest 38% of all incremental training iterations, the value IEC remains < 50, consequently reducing the overall impact of IEC values.

Average accuracies for old classes in all the validation sets for incrementally learning models trained via the least-frequently-occurring classes first are given in Table 14. The best-case accuracy of 61.71% is achieved when new classes are added in batches of NC=1 with an IEC value of 150. Just like for all classes, ANOVA tests show that there is a statistically significant difference in the average accuracies for all validation sets when BC and NC vary, but not when IEC varies.

The reason for getting the best results with NC=1 could be that in the case of training with least-frequent classes first, during each iteration, the number of instances in old classes is less compared to instances in new classes. When new classes are added in larger batch size, the dataset becomes more imbalanced, compared to when new classes are added in smaller batches. Thus, with large values of NC, the model becomes biased toward new classes, resulting in poor performance for old classes (catastrophic forgetting).

Average accuracies for new classes in all the validation sets for incrementally learning models is depicted in Table 15. Best case accuracy of 73.61% is achieved when new classes are added in batches of 5 (BC = 5, NC = 5) with an IEC value of 50.

Unlike the results obtained for all classes and old classes, ANOVA tests show a statistically significant difference in the average accuracies for all validation sets when BC and NC vary ($p - value = 0.0018$), and when IEC varies ($p - value = 0.014$).

The reason for getting better results with bigger batch sizes could be that, when training the model with least-frequently-occurring classes first, new classes in each iteration contain

Table 15 Average of percentage accuracies for private dataset for new classes in all batches of validation sets when incremental learning is performed using least frequently occurring classes first

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	71.94 (0.78)	69.09 (0.68)	70.68 (0.69)	71.68 (0.69)	71.73 (0.68)
Exp2	73.09 (0.71)	69.30 (0.68)	69.50 (0.68)	69.09 (0.68)	69.04 (0.68)
Exp3	68.68 (0.68)	67.44 (0.66)	67.76 (0.66)	65.64 (0.64)	65.74 (0.64)
Exp4	74.28 (0.73)	71.52 (0.70)	70.24 (0.69)	72.29 (0.71)	72.25 (0.71)
Exp5	**73.61** (0.72)	68.62 (0.67)	67.78 (0.66)	68.73 (0.67)	69.81 (0.68)

Weighted F1 in brackets. Bold indicates highest values

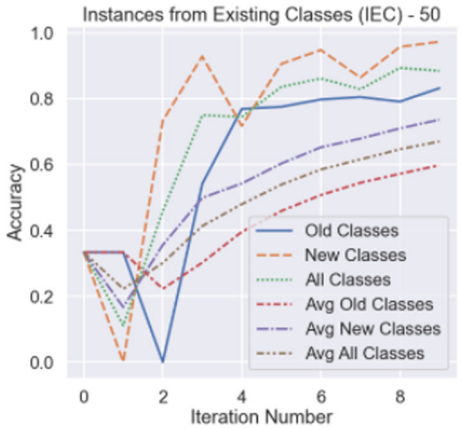

Fig. 8 Comparison of accuracies for old, new, and all classes for Exp5 with IEC = 50, for different iterations of incremental models trained on private dataset using least-frequently-occurring classes first

the majority of instances, hence adding a larger batch of new classes feed the model with more instances from new classes, resulting in improved performance for the new classes (but catastrophic forgetting for old classes, as discussed above).

The overall results shown in Fig. 8 highlight that, for each iteration in Exp5 (BC=5, NC=5) with IEC value of 50, the accuracy values for newly added classes are higher compared to old or existing classes. The reason for such behaviour could be the fact that since models are trained via least-frequently-occurring classes first, new classes in each iteration contain more instances compared to old classes. Hence the model learns to classify new classes in a more accurate manner compared to old classes which are least represented within a training batch. Once again, this is the opposite of what is observed when the incremental learning models are trained via most-frequently-occurring classes first as depicted in Fig. 7.

Another important observation is that, for early batches, the accuracy for both new and old classes is lower compared to the later batches. This behaviour can be the result of the class order, since the earlier iterations contain classes with fewer instances, resulting in models having less information to learn in the early batches as compared to later batches.

Similar observations are made for different values of BC and NC as shown in Appendix B[4]

4.2.3 Private dataset–random addition of new classes

The results for the experiments for our private dataset where new classes are added in random order are presented in Tables 16, 17, and 18.

Table 16 shows that for all the classes the average best case accuracy of 95.25% is achieved for all validation sets in case of Exp1 (BC = 2 and NC = 1) when the IEC value is 250. The result shows an improvement of 1.55% over the baseline results.

Just like when the most frequent classes are learned first, ANOVA and pairwise *T*-tests show that there is no statistical difference between the average accuracies of "Exp1"–"Exp5" and between IEC=200 and IEC values in the list {150, 200, 250}.

For the average accuracy of validation sets for old classes in the private dataset, the results are presented in Table 17. The result shows that the highest average accuracy of 95.32% is achieved for Exp1 (BC =2, NC =1) with 250 instances from existing classes (IEC). ANOVA and pairwise T-tests show that there is no statistical difference between the average accuracies of "Exp1"–"Exp5" and between IEC=250 and IEC values in the list 150, 200, 250.

The reason for the high performance for Exp1 when NC=1 could be the fact that in each iteration, a majority of classes are from the previous batches that are already used to train the model in the previous iterations. Only one new class is added in each batch. Hence the model returns the best performance in the case of Exp1 for old classes.

Table 18 depicts the results for the average accuracies for new classes for all the validation sets. The results show that the best-case average accuracy of 91.66% is achieved for Exp5 (BC = 5, NC = 5) with the IEC value of 250. The reason for this behaviour could be that, when new classes are added in larger batches, the model has more information to learn compared to when new classes are added in smaller batches.

[4] Link to the code and appendix: http://bit.ly/3hC6ved.

Table 16 Average of percentage accuracies for private dataset for all batches of validation sets when incremental learning is performed by randomly adding new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	94.02 (0.93)	94.56 (0.93)	94.24 (0.93)	94.15 (0.93)	**95.25** (0.94)
Exp2	92.07 (0.91)	93.56 (0.92)	94.18 (0.93)	94.01 (0.93)	94.21 (0.92)
Exp3	93.09 (0.91)	93.35 (0.92)	94.18 (0.92)	94.52 (0.93)	94.74 (0.93)
Exp4	92.55 (0.90)	93.62 (0.91)	93.90 (0.92)	93.32 (0.94)	93.62 (0.92)
Exp5	92.59 (0.91)	92.34 (0.90)	93.45 (0.92)	95.16 (0.94)	93.97 (0.92)
Baseline [50]			93.77		

Weighted F1 in brackets. Bold indicates highest values

Table 17 Average of percentage accuracies for private dataset for old classes in all batches of validation sets when incremental learning is performed by random addition of new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	93.88 (0.92)	94.58 (0.93)	94.29 (0.94)	94.16 (0.93)	**95.32** (0.94)
Exp2	91.82 (0.90)	93.52 (0.93)	94.24 (0.93)	94.15 (0.94)	94.28 (0.93)
Exp3	93.01 (0.92)	92.96 (0.93)	94.42 (0.92)	94.50 (0.94)	95.10 (0.94)
Exp4	92.17 (0.91)	93.62 (0.92)	94.01 (0.93)	93.30 (0.92)	93.99 (0.93)
Exp5	91.80 (0.90)	91.65 (0.90)	92.92 (0.91)	95.25 (0.94)	93.34 (0.92)

Weighted F1 in brackets. Bold indicates the highest values

Table 18 Average of percentage accuracies for private dataset for new classes in all batches of validation sets when incremental learning is performed by randomly adding new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	74.50 (0.72)	76.08 (0.75)	79.28 (0.78)	76.59 (0.75)	78.81 (0.78)
Exp2	77.03 (0.76)	83.18 (0.82)	80.67 (0.80)	83.74 (0.83)	81.55 (0.80)
Exp3	81.38 (0.80)	88.84 (0.87)	82.21 (0.81)	89.56 (0.87)	83.17 (0.81)
Exp4	81.08 (0.80)	82.86 (0.80)	81.86 (0.80)	83.12 (0.81)	82.19 (0.80)
Exp5	84.56 (0.81)	84.99 (0.82)	90.20 (0.88)	85.92 (0.85)	**91.66 (0.90)**

Weighted F1 in brackets. Bold indicates the highest values

However, for the random addition of new classes, the average best-case accuracy for new classes (91.66%) is still less than the best-case accuracy for old classes (95.32%). This behaviour can be attributed to the nature of the incremental learning model where the difference between the number of old and new classes is not very high for initial training iterations. However, for later training iterations, this difference becomes higher as the incremental model is repeatedly trained on old classes for more iterations compared to new classes.

In this specific case, ANOVA tests reveal that there is no statistically significant effect of parameters IEC, but there is a significant effect of parameters BC, NC on the average accuracy ($p - value = 0.061$ and $p - value = 0.00009$ respectively). However, the statistical significance for new classes in the case of various experiments in this section cannot be considered as definitely conclusive. Indeed, the order in which the classes have been added is totally random, and, since the dataset is imbalanced, the manner in which the documents arrive may change the effect of IEC, BC, and NC values (as shown in the previous experiments with the extreme cases of most-frequently-occurring classes first and least-frequently-occurring classes first).

The overall results from this section, detailed in Appendix C,[5] show that in case of random addition of new classes, the maximum IEC value of 250 produces the highest average accuracies for respectively all, new and old classes during all training iterations. This result affirms the hypothesis that overall, using more instances from previous iterations can improve model performance in case of random addition of new classes. However, the significance tests show that the performance difference between the IEC values of 150, 200, and 250 is not statistically significant, hence to save training time and memory, the IEC value of 150 may be used for incremental training.

[5] Link to the appendix: http://bit.ly/3hC6ved.

Table 19 Average of percentage accuracies for RVL-CDIP dataset for all batches of validation sets when incremental learning is performed by randomly adding new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	74.86 (0.77)	81.32 (0.79)	81.99 (0.79)	83.31 (0.81)	**84.36** (0.83)
Exp2	73.83 (0.71)	80.51 (0.79)	83.16 (0.82)	84.07 (0.82)	83.23 (0.82)
Exp3	79.07 (0.77)	79.94 (0.77)	81.88 (0.80)	81.97 (0.80)	81.69 (0.80)
Exp4	71.16 (0.70)	80.51 (0.79)	79.73 (0.78)	80.75 (0.79)	83.93 (0.79)
Exp5	74.47 (0.73)	79.98 (0.78)	80.07 (0.78)	82.18 (0.80)	80.33 (0.79)
Baseline [50]			80.76		

Weighted F1 in brackets. Bold indicates the highest values

Table 20 Average of percentage accuracies for RVL-CDIP dataset for old classes in all batches of validation sets when incremental learning is performed by randomly adding new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	71.82 (0.70)	79.60 (0.78)	80.87 (0.79)	82.68 (0.80)	**83.00** (0.81)
Exp2	67.70 (0.65)	76.95 (0.75)	80.77 (0.79)	81.71 (0.80)	81.08 (0.80)
Exp3	74.67 (0.73)	77.03 (0.76)	80.12 (0.80)	80.63 (0.79)	80.44 (0.78)
Exp4	61.12 (0.60)	75.20 (0.74)	75.95 (0.75)	77.09 (0.78)	82.36 (0.80)
Exp5	67.43 (0.66)	76.45 (0.75)	76.91 (0.76)	80.62 (0.79)	78.93 (0.77)

Weighted F1 in brackets. Bold indicates the highest values

Table 21 Average of percentage accuracies for RVL-CDIP dataset for new classes in all batches of validation sets when incremental learning is performed by randomly adding new classes

Exp.Id	IEC values				
	50	100	150	200	250
Exp1	**93.71** (0.92)	92.43 (0.91)	90.45 (0.89)	88.76 (0.87)	86.91 (0.85)
Exp2	92.96 (0.91)	91.22 (0.90)	90.66 (0.89)	91.16 (0.90)	88.91 (0.87)
Exp3	90.13 (0.89)	89.02 (0.88)	87.55 (0.86)	85.26 (0.84)	85.83 (0.83)
Exp4	89.87 (0.86)	89.70 (0.86)	86.16 (0.85)	88.33 (0.86)	86.73 (0.85)
Exp5	90.90 (0.89)	88.27 (0.86)	87.24 (0.86)	86.92 (0.85)	85.31 (0.84)

Weighted F1 in brackets. Bold indicates the highest values

4.3 Effects of BC, NC, and IEC values for RVL-CDIP Dataset

Our private real-world dataset is highly imbalanced, hence we evaluated three training scenarios depending on the order of addition of new classes in each incremental learning iteration (most-frequently-occurring classes first, least-frequently-occurring classes first and random. On the contrary, the publicly available RVL-CDIP dataset is balanced. Hence experiments are only performed where new classes are randomly added for each incremental learning iteration. In this section, we will analyse the differences observed when comparing random addition of new classes using a balanced dataset on one hand (RVL-CDIP) and an imbalanced dataset (our private dataset, see Sect. 4.2.3).

The results for the experiments for RVL-CDIP dataset where new classes are added in random order are presented in Tables 19, 20, and 21.

Table 19 depicts the results for the average accuracies for all classes in all the validation sets. Just like in the imbalanced case, the best average accuracy (here 84.36%) is achieved for Exp1 (BC = 2, NC = 1) with IEC = 250. Thus, it seems that, independently of the balanced or imbalanced nature of the dataset, when new classes are added in a random order, adding as few new classes as possible in each iteration, with a large memory size, is the best strategy. The result further depicts an improvement of 3.6% over the baseline results.

However, we must moderate this conclusion by noting that, here again, ANOVA and pairwise T-tests show that there is no statistical difference between the average accuracies of "Exp1"–"Exp5" and between IEC=200 and IEC values in the list {150, 200, 250}. Thus, to save training time and memory, the IEC value of 150 may be used for incremental training.

Table 20 depicts the results for the average accuracies for old classes only for all the validation sets. Again, like in the imbalanced case, the results show that the best-case average accuracy (here 83.00%) is achieved for Exp1 (BC = 2, NC = 1) with IEC=250. The analysis of statistical significance using ANOVA and pairwise T-tests is also consistent with

Table 22 Results for the comparison of incremental learning model with static learning model

	Private dataset	RVL-CDIP
Best-case IEC	IEC = 200	IEC = 200
Statistical significance for IEC values	Two groups: (50, 100) and (150, 200,250). No significant performance difference for IEC values within groups but significant for IEC values between groups	Two groups: (50, 100) and (150, 200, 250). No significant performance difference for IEC values within groups but significant for IEC values between groups
Best case BC, NC values	Exp5 (BC = 2, NC = 2)	Exp1 (BC = 5, NC = 5)
Statistical significance for BC, NC values	Difference in results not statistically significant for various BC and NC values	Difference in results not statistically significant for various BC and NC values

IEC instances from existing classes, *BC* number of base classes and, *NC* number of new classes

the result obtained for our private dataset for the same experiment.

Finally, Table 21 depicts the results for the average accuracies for new classes only for all the validation sets. The results show that the best-case average accuracy of 93.77% is achieved for Exp1 (BC = 2, NC = 1) with the IEC value of 50. This behaviour is different than the behaviour observed on our imbalanced dataset, where the best-case accuracy for new classes was achieved for Exp5 (BC = 5, NC = 5) with the IEC value of 250. The reason for this difference could be associated with the nature of the data. Indeed, with our highly imbalanced private dataset, when new classes are added in a random order, it can happen that the instances added for some of the new classes are fewer than the total number of instances in the memory. Hence, in the imbalanced case, the new classes are better recognized when we add more new classes within one iteration, whereas when the dataset is balanced, it is still best to add as few new classes within one iteration as possible. To counterbalance the small number of new classes and encourage the plasticity of the model, new classes are learned best when we keep only a small size of memory (IEC=50).

In this case, ANOVA tests and pairwise T-tests show that the effect of the number of base classes (BC) and new classes (NC) is statistically significant (here $p - value = 0.03$), and that an IEC value of 50 gives significantly better performances than other IEC values.

The figures containing detailed results for the RVL-CDIP, where classes are randomly added for each new iteration are presented in Appendix D.[6]

Even though the following findings should be confirmed with experiments on other balanced datasets, the results presented in this section tend to show that, when the dataset is balanced, the best performances are achieved when we add as few classes as possible in each iteration. As for the number of instances from existing classes (and thus the memory size), it can be fixed by finding a good tradeoff between plasticity, stability, and training time/memory (here IEC=150).

5 Discussion & recommendations

The findings of this study demonstrate that the rehearsal-based incremental learning strategy presented herein surpasses the baseline model's performance by 1.55% and 3.66% on our private dataset and RVL-CDIP datasets, respectively. This superior performance can be attributed to the effective utilization of weight sharing among transformer models across multiple iterations, enabling the model to effectively retain the knowledge of previously encountered classes while learning new ones.

Furthermore, the comparison of the rehearsal-based incremental learning model with static model (on a static test dataset) reveals some interesting observations. Table 22 summarizes these observations, for both our imbalanced private dataset and the balanced RVL-CDIP dataset.

For both datasets, the optimal value of parameter IEC (number of instances from existing classes kept in memory at each incremental learning iteration), which gives the best average accuracy, is IEC=200. However, pairwise T-tests show that based on statistical significance, IEC values can be divided into two groups: (50 and 100), and (150, 200, 250). There is no statistically significant performance difference within groups, but a significant performance difference between groups. Thus, we can recommend using an IEC value of 150, so as to get results comparable with IEC = 200, yet saving training memory and resources.

For base class (BC) and new class values (NC), even though statistical significance tests are not conclusive, we can recommend using a small number of new classes in real-life scenarios (NC=1 or NC=2 if possible), since real-life datasets are mostly imbalanced, and adding too many new

[6] Link to the appendix: http://bit.ly/3hC6ved.

Table 23 Overall results for experiments to find best values, for instance, from existing classes (IEC), number of base classes (BC), and number of new classes (NC) for incremental learning model for multiple iterations

	Private dataset			RVL-CDIP dataset
	MFO	LFO	RCA	RCA
Best-case IEC	IEC = 200	IEC = 50	IEC = 250	IEC = 250
Statistical significance for IEC values	Two groups: (50, 100) and (150, 200, 250). No significant performance difference for IEC values within groups, but significant for IEC values between groups	No statistically significant difference observed for various IEC values	Two groups: (50, 100) and (150, 200, 250). No significant performance difference for IEC values within groups, but significant for IEC values between groups	Two groups: (50, 100) and (150, 200, 250). No significant performance difference for IEC values within groups, but significant for IEC values between groups
Best case BC, NC values	BC = 2, NC = 1	BC = 5, NC = 5	BC = 2, NC = 1	BC = 2, NC = 1
Statistical significance for BC, NC values	Difference in results not statistically significant for various BC and NC values	Two groups: (Exp1, Exp2, Exp3) and (Exp4, Exp5). Performance difference not significant within groups but significant between groups	Difference in results not statistically significant for various BC and NC values	Difference in results not statistically significant for various BC and NC values
Better average accuracy	Old classes	New classes	Old classes	New classes

MFO training with most-frequently occurring classes, *LFO* training with least-frequently occurring classes, *RCA* training with random class addition

classes at each iteration may increase the overall class imbalance.

A summary of our analysis of the effects on incrementally trained models of (i) the order in which new classes are added, (ii) the IEC (instances from existing classes), and (iii) the batch size of the base and new classes (BC/NC), for both our private (imbalanced) dataset and the RVL-CDIP dataset is presented in Table 23.

Overall, based on our experimental results, it can be concluded that the choice of the number of instances from existing classes (IEC), and the batch size of the base and new classes (BC and NC) depends upon the nature of the dataset type (balanced or imbalanced). In the presence of an imbalanced dataset, the order (most/least frequently or random selection) in which the classes are selected for incremental training in multiple iterations, also affects the choice of values for IEC, BC, and NC.

In most real-life scenarios, the datasets are imbalanced and the models are trained with most-frequently-occurring classes first, or random class addition (the scenario of least-occurring classes first being very unlikely to happen in real applications). In such cases, or even if the expected number of instances from future classes is unknown, we recommend using 150 instances from existing classes (IEC) per iteration, while the number of base classes (BC) is kept to minimum and new classes are added one by one (NC=1).

However, if the dataset is initially very small, with very few instances per base class, and a large number of incoming documents from new classes are expected over time, then we recommend using a larger batch size for new classes (NC).

6 Conclusion and future work

Document classification is an important task for corporate organizations and private companies. However, the incessant evolutions in the documents to classify, and particularly the arrival of new classes of documents over time, makes it difficult to train a static machine learning model for document classification. Incremental learning models cater to such scenarios where document datasets are constantly evolving with new information.

In this study, we propose a rehearsal-based incremental learning model for document classification that learns from text extracted from evolving document datasets. The performance of the proposed model is compared with traditional deep learning models for the classification of business documents using textual information from our private and RVL-CDIP datasets. The results show that rehearsal-based incremental learning models if trained using all the training data (even in multiple iterations), performs similar to static machine learning model on a fixed test set. This result shows that an incremental learning model can be successfully

employed for document classification in cases where data is evolving at run-time with the arrival of new documents since the results are similar to those obtained using a static learning model.

Moreover, we investigate the optimal values of the parameters controlling the memory size/variety (namely parameters BC and IEC), and the batch size for new classes (NC), and analyzed their effects on the performances of the rehearsal-based model along its training iterations. Based on this analysis, we formulate a list of recommendations for these values, depending on the scenario at hand and the nature of the dataset.

Though this research work analyses various aspects of incremental learning model for document classification, some possibilities are yet to be explored.

First, we could make our evaluation scenarios even more realistic, for instance by simulating scenarios where the instances from NC new classes are not all fed to the model in only one training iteration, or the value of parameter BC differs largely from NC, or the value of IEC can vary across training iterations.

Second, this study explores various aspects of the rehearsal-based approach for incremental learning and its comparison with a traditional deep learning model for document classification. It can be useful to explore how other incremental learning approaches such as regularization-based approaches and variational continual learning can be used for document classification, and how they compare with static deep learning models.

Finally, another very interesting future direction is to use not only text but also layout (image) information from the document, for multimodal incremental document classification.

Funding This work is supported by the Region Nouvelle Aquitaine under the grant number 2019-1R50120 (CRASD project) and AAPR2020-2019-8496610 (CRASD2 project) and by the LabCom IDEAS under the Grant Number ANR-18-LCV3-0008.

Declarations

Conflict of interest The authors have no financial or Conflict of interest to declare that are relevant to the content of this article.

References

1. Asim, M.N., Khan, M.U.G., Malik, M.I., Dengel, A., Ahmed, S.: A robust hybrid approach for textual document classification. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1390–1396, IEEE (2019)
2. Kölsch, A., Afzal, M.Z., Ebbecke, M., Liwicki, M.: Real-time document image classification using deep CNN and extreme learning machines. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1318–1323, IEEE (2017)
3. Gallo, I., Calefati, A., Nawaz, S., Janjua, M.K.: Image and encoded text fusion for multi-modal classification. In: 2018 Digital Image Computing: Techniques and Applications (DICTA), pp. 1–7, IEEE (2018)
4. Rebuffi, S.-A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2001–2010 (2017)
5. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 139–154 (2018)
6. Cote, M., Albu, A.B.: Texture sparseness for pixel classification of business document images. Int. J. Doc. Anal. Recognit. (IJDAR) **17**(3), 257–273 (2014)
7. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 374–382 (2019)
8. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**(13), 3521–3526 (2017)
9. D'Andecy, V.P., Joseph, A., Ogier, J.-M.: Indus: incremental document understanding system focus on document classification. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 239–244, IEEE (2018)
10. Alhaj, Y.A., Xiang, J., Zhao, D., Al-Qaness, M.A., Abd Elaziz, M., Dahou, A.: A study of the effects of stemming strategies on Arabic document classification. IEEE Access **7**, 32664–32671 (2019)
11. Shahkolaei, A., Beghdadi, A., Cheriet, M.: Blind quality assessment metric and degradation classification for degraded document images. Signal Process. Image Commun. **76**, 11–21 (2019)
12. Sanchez-Pi, N., Martí, L., Garcia, A.C.B.: Improving ontology-based text classification: an occupational health and security application. J. Appl. Log. **17**, 48–58 (2016)
13. Gayathri, M., Kannan, R.J.: Ontology based concept extraction and classification of ayurvedic documents. Procedia Comput. Sci. **172**, 511–516 (2020)
14. Walczak, S., Kellogg, D.L.: A heuristic text analytic approach for classifying research articles. Intell. Inf. Manag. **7**(1), 7 (2015)
15. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation. arXiv preprint arXiv:2010.15277 (2020)
16. Luo, Y., Yin, L., Bai, W., Mao, K.: An appraisal of incremental learning methods. Entropy **22**(11), 1190 (2020)
17. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint arXiv:1904.07734 (2019)
18. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 532–547 (2018)
19. Mermillod, M., Bugaiska, A., Bonin, P.: The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. Front. Psychol. **4**, 504 (2013)
20. Zeng, G., Chen, Y., Cui, B., Yu, S.: Continual learning of context-dependent processing in neural networks. Nat. Mach. Intell. **1**(8), 364–372 (2019)
21. Farajtabar, M., Azizan, N., Mott, A., Li, A.: Orthogonal gradient descent for continual learning. In: International Conference on Artificial Intelligence and Statistics, pp. 3762–3773, PMLR (2020)
22. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 233–248 (2018)

23. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
24. Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L., Zhang, H., Kuo, C.-C.J.: Class-incremental learning via deep model consolidation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1131–1140 (2020)
25. Lee, K., Lee, K., Shin, J., Lee, H.: Overcoming catastrophic forgetting with unlabeled data in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 312–321 (2019)
26. Roy, D., Panda, P., Roy, K.: Tree-CNN: a hierarchical deep convolutional neural network for incremental learning. Neural Netw. **121**, 148–160 (2020)
27. Mandivarapu, J.K., Camp, B., Estrada, R.: Self-net: lifelong learning via continual self-modeling. Front. Artif. Intell. **3**, 19 (2020)
28. Polikar, R., Upda, L., Upda, S.S., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. IEEE Trans. Syst. Man, Cybern. Part C (Appl. Rev.) **31**(4), 497–508 (2001)
29. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
30. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547 (2017)
31. Nguyen, C.V., Li, Y., Bui, T.D., Turner, R.E.: Variational continual learning. arXiv preprint arXiv:1710.10628 (2017)
32. Farquhar, S., Gal, Y.: A unifying Bayesian view of continual learning. arXiv preprint arXiv:1902.06494 (2019)
33. Chen, Y., Diethe, T., Lawrence, N.: Facilitating bayesian continual learning by natural gradients and stein gradients. In: Continual Learning Workshop of 32nd Conference on Neural Information Processing Systems (NeurIPS 2018) (2019)
34. Adel, T., Zhao, H., Turner, R.E.: Continual learning with adaptive weights (claw). In: International Conference on Learning Representations (2019)
35. Ebrahimi, S., Elhoseiny, M., Darrell, T., Rohrbach, M.: Uncertainty-guided continual learning with Bayesian neural networks. arXiv preprint arXiv:1906.02425 (2019)
36. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Zhang, Z., Fu, Y.: Incremental classifier learning with generative adversarial networks. arXiv preprint arXiv:1802.00853 (2018)
37. Masarczyk, W., Tautkute, I.: Reducing catastrophic forgetting with learning on synthetic data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 252–253 (2020)
38. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. arXiv preprint arXiv:1705.08690 (2017)
39. Kemker, R., Kanan, C.: Fearnet: brain-inspired model for incremental learning. arXiv preprint arXiv:1711.10563 (2017)
40. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 831–839 (2019)
41. Arif-Uz-Zaman, K., Cholette, M.E., Ma, L., Karim, A.: Extracting failure time data from industrial maintenance records using text mining. Adv. Eng. Inform. **33**, 388–396 (2017)
42. Mahamoud, I.S., Voerman, J., Coustaty, M., Joseph, A., d'Andecy, V.P., Ogier, J.-M.: Multimodal attention-based learning for imbalanced corporate documents classification. In: International Conference on Document Analysis and Recognition, pp. 223–237, Springer (2021)
43. Choe, J.-W., Lee, D.-G.: Trends 21 corpus: public web resources and search tools. Stud. Korean Cult. **64**, 1–20 (2014)
44. Hitt, M.A., Ireland, R.D., Hoskisson, R.E.: Strategic management: concepts and cases: competitiveness and globalization. Cengage Learn. (2016)
45. Ahn, H., Cha, S., Lee, D., Moon, T.: Uncertainty-based continual learning with adaptive regularization. arXiv preprint arXiv:1905.11614 (2019)
46. Shan, G., Xu, S., Yang, L., Jia, S., Xiang, Y.: Learn#: a novel incremental learning method for text classification. Expert Syst. Appl. **147**, 113198 (2020)
47. Madhusudhanan, S., Jaganathan, S., LS, J.: Incremental learning for classification of unstructured data using extreme learning machine. Algorithms **11**(10), 158 (2018)
48. Jang, J., Kim, Y., Choi, K., Suh, S.: Sequential targeting: an incremental learning approach for data imbalance in text classification. arXiv preprint arXiv:2011.10216, (2020)
49. Xia, C., Yin, W., Feng, Y., Yu, P.: Incremental few-shot text classification with multi-round new classes: formulation, dataset and system. arXiv preprint arXiv:2104.11882 (2021)
50. Voerman, J., Mahamoud, I.S., Joseph, A., Coustaty, M., D'Andecy, V.P., Ogier, J.-M.: Toward an incremental classification process of document stream using a cascade of systems. In: Document Analysis and Recognition–ICDAR 2021 Workshops: Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16, pp. 240–254, Springer (2021)
51. Ravuri, S., Vinyals, O.: Classification accuracy score for conditional generative models. arXiv preprint arXiv:1905.10887 (2019)
52. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 991–995, IEEE (2015)
53. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
54. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
55. Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., Schwab, D.: Flaubert: unsupervised language model pre-training for french. arXiv preprint arXiv:1912.05372 (2019)
56. Emerson, R.W.: Anova and t-tests. J. Vis. Impair. Blind. **111**(2), 193–196 (2017)