



The class imbalance problem in deep learning

Kushankur Ghosh¹ · Colin Bellinger² · Roberto Corizzo³ · Paula Branco⁴
Bartosz Krawczyk⁵ · Nathalie Japkowicz³

Received: 2 April 2022 / Revised: 12 August 2022 / Accepted: 2 October 2022 /

Published online: 28 December 2022

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2022

Abstract

Deep learning has recently unleashed the ability for Machine learning (ML) to make unparalleled strides. It did so by confronting and successfully addressing, at least to a certain extent, the knowledge bottleneck that paralyzed ML and artificial intelligence for decades. The community is currently basking in deep learning's success, but a question that comes to mind is: have all of the issues previously affecting machine learning systems been solved by deep learning or do some issues remain for which deep learning is not a bulletproof solution? This question in the context of the class imbalance becomes a motivation for this paper. Imbalance problem was first recognized almost three decades ago and has remained a critical challenge at least for traditional learning approaches. Our goal is to investigate whether the tight dependency between class imbalances, concept complexities, dataset size and classifier performance, known to exist in traditional learning systems, is alleviated in any way in deep learning approaches and to what extent, if any, network depth and regularization can help. To answer these questions we conduct a survey of the recent literature focused on deep learning and the class imbalance problem as well as a series of controlled experiments on both artificial and real-world domains. This allows us to formulate lessons learned about the impact of class imbalance on deep learning models, as well as pose open challenges that should be tackled by researchers in this field.

Keywords Deep learning · Class imbalance · Concept complexity

1 Introduction

The purpose of this article is to align the progress made on the deep learning front with one of the main questions that has been debated in the traditional machine learning literature for the past three decades: the class imbalance problem (Fernández et al., 2018). As is well-known or as new practitioners of machine learning and deep learning methods soon discover in their investigations, the class imbalance problem is a very

Editors: Nuno Moniz, Luís Torgo, Michal Wozniak, Shuo Wang.

✉ Paula Branco
pbranco@uottawa.ca

Extended author information available on the last page of the article

prevalent problem that is hard to address (Branco et al., 2016). In a binary classification setting, there is a class imbalance if one class is represented by the majority of instances present in the data set and the other one is represented by only a minority of instances. Class imbalances extend to and is, in fact, magnified in multi-class, multi-label, multi-instance learning as well as in regression problems and so forth (Krawczyk, 2016).

Research goal To propose a thorough investigation of the effects of class imbalance on deep learning models, understand the interplay between deep learning architectures and skewed data, as well as provide insights into how imbalanced data affect deep models differently from their shallow counterparts.

Motivation The class imbalance problem is prevalent in real-world problems, especially in the ones tackled by deep learning models. However, so far, research aimed at making deep neural networks more robust to class imbalances mostly mimics popular approaches for shallow models. In other words, there is a lack of holistic work that offers an understanding of how class imbalances impact deep learning models, and how their performance is tied to various other data-level difficulties present in skewed data. It is our belief that one cannot simply transfer the solutions proposed for shallow models to deep ones, as that does not take into consideration the unique nature of the problem. Therefore, there is a need for an in-depth analysis of how deep neural networks are impacted by various problems coming from imbalanced data.

Overview This article attempts to answer questions about class imbalances in deep learning through both a survey of the literature and a series of controlled experiments. At the center of the investigation is the question of whether the response of deep learning models is similar or different from that of traditional learners, and whether adding depth to neural networks helps, hinders, or has no effect on the class imbalance problem. The work presented in this article expands the work previously reported in (Ghosh et al., 2021).

Contributions This manuscript offers the following research contributions:

- *Thorough literature survey* We present a two-part overview of the most important and impactful works dealing with the class imbalance problem in deep learning. In the first part we discuss existing works that investigate the impact of class imbalance on deep learning architectures. In the second part, we create a taxonomy of existing deep approaches that tackle class imbalances and discuss the previously proposed distinct groups of mechanisms that offer skew-insensitiveness to deep neural networks.
- *Understanding the impact of class imbalance on deep learning models* We propose to understand whether the impact of class imbalance on deep learning models is similar to its impact on their shallow learning counterparts. We extract the factors unique to deep architectures that appear to play a significant role in the class imbalance setting.
- *Investigating interplay between the depth of neural networks and class imbalance* We analyze whether there are trends in the relationship between the depth of deep neural networks and their robustness to various levels of class imbalance, data scarcity and data complexity.
- *Examining the role of regularization approaches in learning from imbalanced data* Finally, we investigate the impact of regularization approaches on the robustness of deep models in skewed learning scenarios.
- *In-depth experimental study* We answer the research questions posed in this manuscript based on a thorough and holistic experimental study carried out on artificial and real datasets coming from diverse domains.

Organization The remainder of the article is divided into ten sections. Section 2 reviews the literature pertaining to the effect of class imbalances on deep learning systems. Since little has been said, so far, on that question, the next six sections seek to answer it more fully in the binary classification setting. Section 3 describes the set-up of the controlled experiments. Section 4 describes the analytical framework followed in discussing the results. Section 5 describes the main takeaways of our experiments. Section 6 lists some notational details useful to keep in mind when reading the next two sections. Sections 7 and 8 dive into the actual results and their analysis for Multi-Layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs), respectively. Section 9 is a second literature review that surveys what has been done in deep learning, so far, to handle the class imbalance problem. Section 10 lists a series of open challenges for class imbalance in deep learning. Section 11 concludes the article. An appendix is also included with additional results that would have taken too much space in the main body of the article.

2 Literature review I: the effect of class imbalances on deep learning systems

2.1 Deep learning background

Deep learning is a branch of machine learning that uses Artificial Neural Networks (ANNs), learning algorithms inspired by the human brain, that seek to approximate an unknown function that maps input data into a target variable value or class. ANNs are composed of a set of weighted interconnected artificial neurons, the basic computational unit of ANNs. Each neuron uses an activation function to transform the weighted inputs into a single output.

The Multi-Layer Perceptron (MLP) is a fully-connected feedforward neural network that contains an input layer, and output layers and at least one hidden layer. We have a fully-connected network when all connections between the different layers are present and we say it is feedforward when the data propagates from the input layer, to each subsequent hidden layer until the output layer in a forward pass. MLPs can be shallow or deep depending on the number of hidden layers that they contain.

Convolutional Neural Networks (CNNs) are feedforward networks that have similarities with the human visual processing system, being highly optimized for processing multidimensional data such as 2d or 3D images. When CNNs were proposed at the end of the 1980's, they were not widely used due restrictions in computational hardware that limited the training of these networks. However, the successful application of a gradient-based learning algorithm to CNNs in 1998 (LeCun et al., 1998) boosted the popularity of this solution. A CNN architecture typically includes multiple alternating convolution and pooling layers and fully-connected layers. The outputs of the convolution and pooling layers are aggregated into a plane, named a feature map. Each node in a given plane is obtained from small regions of the connected planes in the previous layers, where a convolutional layer learns to detect multiple features and the pooling layer merges similar features acting as a dimensionality reduction (LeCun et al., 2015). After the convolutional and pooling layers the output is flattened and the fully-connected layers are applied.

With the growing use of deep learning and its successful application in the particular case of images, several CNN architectures have become popular. Some well-known examples of specific CNN architectures include: LeNet-5 (LeCun et al., 1998), ResNet-10 (Simon

Table 1 Main publications exploring the impact of the class imbalance problem in the context of neural networks

References	Architecture	Datasets	Task
Anand et al. (1993)	MLP	Synthetic dataset; speech recognition; binarized iris	Binary
Murphrey et al. (2004)	MLP, RBF, ART	Data collected from 3 vehicle models in an automobile assembly plant	Binary
Hensman and Masko (2015)	CNN	CIFAR-10 (11 versions)	Multiclass
Lee et al. (2016)	CNN	WHOI-Plankton	Multiclass
Wang et al. (2016)	MLP	CIFAR-100; 20 Newsgroups	Binary
Raj et al. (2016)	MC-DCNN	XING, Earthquake, ECG	Binary
Ding et al. (2017)	CNN	EmotioNet 2017	Binary
Pulgar et al. (2017)	CNN	Traffic signal dataset	Multiclass
Pouyanfar et al. (2018)	CNN	Images from network cameras	Multiclass
Buda et al. (2018)	CNN	MNIST (81 versions); CIFAR-10 (16 versions); ImageNet (3 versions)	Multiclass
Ya-Guan et al. (2020)	MLP, CNN	HTRU2, MNIST, Seismic-bumps	Binary and Multiclass
Johnson and Khoshgoftaar (2020)	MLP	2 Medicare datasets	Binary
Valova et al. (2020)	CNN	Architecture heritage elements; Boats	Multiclass
Bria et al. (2020)	CNN	Microcalcification; Microaneurysm	Binary

et al., 2016), AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan & Zisserman, 2014), NiN (Lin et al., 2013) or All-CNN (Springenberg et al., 2014).

2.2 Class imbalances and deep learning

Research in deep learning has attracted much attention. However, only few existing works seek to understand the relation between the class imbalance and deep learning methods. Table 1 summarizes the main publications that explore how the class imbalance problem impacts the performance of neural networks. We observe that the majority of works (9 out of 14) are focused on image data and thus take into account CNNs. There are 5 publications that cover the issue using other networks for tabular data and one paper that addresses this problem for time series data. Overall, there is an interest in both binary and multiclass problems on a diversity of datasets. Multiple works carry out a study for a particular application domain that is imbalanced (e.g. anomaly in vehicles data, lesion detection or Plankton data). Few works present a more broad study considering several datasets and multiple imbalanced version and/or imbalance ratios. We also observe a growing trend on the exploration of the class imbalance problem when using deep learning.

Anand et al. (1993) present the first work that investigates the impact of the class imbalance problem in neural networks convergence. The performance of an MLP with one hidden layer was evaluated on 3 datasets. The nodes and learning rate were adapted to each one of the datasets tested. The authors showed that, in an imbalanced setting, there is a critical difference in the lengths of the classes gradient component. The length of minority class gradient is much smaller than that of the majority class, which means that the majority class dominates the weight update process of the model. This leads to a fast reduction of the majority class error in early iterations while having the opposite effect on the minority class, causing a slow convergence phenomenon.

Murphrey et al. (2004) studied the sensitivity to class separability and noise on three neural network architectures: MLP, Radial Basis Function (RBF), and Fuzzy Adaptive Resonance Theory network (ART). The authors used the data obtained from three different vehicle models collected at a test site in an automobile assembly plant. Each network was evaluated with hyperparameters determined for each dataset. In this study it was found that the performance of the neural networks on imbalanced problems is related to the architecture of the network and the separability of the classes. In particular, both the BP and ART achieved a good performance on imbalanced data while the RBF network was not able to learn the minority class features in a satisfactory way. The performance on the majority class was not affected in any of the network architectures.

Hensman and Masko (2015) analysed the impact of the class imbalance problem in a multi-class setting when training a CNN. The network was trained with a balanced and multiple imbalanced distributions generated from the CIFAR-10 dataset (Krizhevsky et al., 2009). The imbalanced distributions generated were as follows: (i) 5 minority and 5 majority classes; (ii) 1 majority and 9 minority classes; (iii) 1 minority and 9 majority classes; (iv) a linear step imbalance; (v) an exponential step imbalance; and (vi) 4 majority and 6 minority classes where similar classes are assigned the same number of examples. For all cases, except (iv) and (v), two imbalance ratios were tested. The balanced distribution exhibited the best overall performance as well as the best minimum performance across all individual classes. A variant of AlexNet (Krizhevsky et al., 2012) with 3 convolutional layers and 10 output nodes was used for all tests. This study showed that imbalanced distributions have a significant impact on the CNN performance. All distributions, with the exception of the two imbalanced distributions containing a single minority class and 9 majority classes, showed a worse overall performance when compared against the balanced distribution with statistical significance. The more skewed the distribution, the worse the performance impact. The two most affected distribution settings were as follows: (i) one majority and 9 minority classes with a higher imbalance ratio; and (ii) the exponential step imbalance. In these scenarios, the CNN exhibits a near zero performance for all classes except one where it exhibits a high performance which suggests the CNN was repeatedly guessing that class. Similar imbalanced distribution settings, containing a less pronounced imbalance yielded a better performance. The impact observed in the performance of the imbalanced distributions with the same number of majority and minority classes was not very strong. Simultaneously, the authors verified that imbalanced distributions with a single minority class were not severely affected, which suggests that the overall performance is more affected when there is only one majority class present.

Lee et al. (2016) studied a particularly imbalanced application by using the WHOI-Plankton (Orenstein et al., 2015) dataset. This dataset contains a total of 103 classes with a single majority class. The five largest classes represent 73.2% 10.6%, 3.5%, 1.9% and 1.3% of the cases. In this difficult scenario, when using a version of the CIFAR-10 network as baseline CNN, the performance achieved on the 5 most frequent classes is high, but it

severely deteriorates for the smallest classes. This confirms the results that were obtained for this dataset by Orenstein et al. (2015). Moreover, it also corroborates the negative impact in the performance associated to the scenario where only one majority class is present as described by Hensman and Masko (2015).

Wang et al. (2016) explore the class imbalance problem in the contexts of images and text documents, focusing on problems related to the loss function used in MLPs. In their experiments, three different imbalance ratios were examined for both contexts. Namely, the ratio between the minority and the majority class cases was set to 20%, 10%, and 5%. The considered baseline model is an MLP trained with mean squared error (MSE) loss for which a performance decrease is observed when the imbalance in the data is more severe. These results are verified for both F1-score and AUC measures and on all imbalanced versions of CIFAR-10 and 20NewsGroup. Wang et al. (2016) show that the MSE loss function is not able to capture the minority class errors when the data has a high imbalance because it is dominated by the large number of majority class cases, a conclusion similar to what was observed by Anand et al. (1993).

In the context of time series data, Raj et al. (2016) studied the use of Multi-Channels Deep Convolution Neural Networks (MC-DCNN) (Zheng et al., 2014) on highly imbalanced domains. The geometric mean (G-Mean) results are highly impacted by the class imbalance remaining zero irrespectively of the number of iterations used. The conclusions of this study are in accordance with studies using other datasets.

Ding et al. (2017) tackles the Facial Action Units (FAUs) recognition, a naturally imbalanced problem. The FAUs recognition is decomposed into 11 binary classification tasks due to the fact that multiple FAUs may be present in a single image. Different CNN architectures are evaluated on the EmotioNet 2017 Challenge Track 1 dataset (Benitez-Quiroz et al., 2017). More precisely, the authors tested one CNN network with 6 layers and 5 networks with a higher number of layers: 4 ResNet with 10, 18, 34 and 50 layers, and a 34-layer non-residual CNN. The authors carried out experiments that focus on the depth rather than width of the networks because of the efficiency of deep neural networks when compared to shallower ones for representing the complex features associated to facial expression recognition (Telgarsky, 2016). The 6-layer CNN produced poor performance on the highly imbalanced classes while the remaining network architectures tested showed a better performance. The networks with 10 or more layers all display a similar overall performance suggesting that using more than 10 layers is not necessary for this problem. The results obtained, when comparing a 6-layer with a 18-layer network, suggest that very deep networks exhibit a faster convergence while achieving better accuracy when compared against the shallower network. However, the authors suggest that the shallower network may continue to converge with the increase of the number of epochs, even though it has a low convergence rate.

Pulgar et al. (2017) carried out a set of experiments using the traffic signal dataset (Stallkamp et al., 2012) to assess the impact of the class imbalance problem on CNNs. The 5 most frequent and the 5 least frequent classes were selected among the 43 classes of the dataset. Four versions of the dataset were then generated: one balanced and 3 with varying imbalance ratio. The imbalanced versions are generated with a ratio between the minority class and majority class cases of 1:10, 1:5 and 1:3. The same CNN architecture is trained on the four dataset variants. The results show that a global good performance is obtained when the classes distribution is balanced. As the imbalance between the classes increases, the performance degrades, which confirms what was also observed in other research works. The authors suggested that the weights calculation carried out in the last fully connected layer may be one of the main aspects affecting the network performance

due to a possible bias towards the majority class. Moreover, the weights calculation at the filters level in the convolutional layers may also be overly biased to the majority classes when a high imbalance is present.

Pouyanfar et al. (2018) used images captured from publicly available network cameras to study the influence of the class imbalance in the performance of a CNN. The VGG-Net (Simonyan & Zisserman, 2014) used on this highly imbalanced multi-class problem produced a poor overall performance and was not able to detect the highly infrequent classes. The authors associated the poor performance observed to the need of using large datasets in order for the CNNs to be able to accurately update the weights.

Buda et al. (2018) studied the performance of different CNN architectures on 3 image datasets (MNIST, CIFAR-10 and ImageNet) of which 100 multi-class imbalanced versions are generated. The modern version of LeNet-5 is used for MNIST (LeCun et al., 1998), All-CNN (Springenberg et al., 2014) is used for CIFAR-10 and the ResNet-10 (Simon et al., 2016) is used for the ImageNet dataset. The parameters are tuned for each dataset and are fixed for all the imbalanced versions generated for a dataset. Two main imbalance types were considered: the step imbalance, where all minority classes have the same number of cases and the same applies to the majority classes, and the linear imbalance, where one minority and one majority class are considered and the remaining classes frequency is obtained by linearly interpolating them. The main results show that both the imbalance ratio and the number of minority classes have a detrimental effect on the classifiers performance. These results confirm the trend previously observed by Hensman and Masko (2015) concerning a higher degradation in the performance for tasks with a higher number of minority classes.

Ya-Guan et al. (2020) evaluated MLPs on binary classification datasets (HTRU2 Lyon et al., 2016 and Seismic-bumps Sikora et al., 2010), and evaluated CNNs on MNIST dataset from which one balanced and three imbalanced multi-class versions were generated. The experiments revealed that on MLPs the error of the majority class decreases as the iterations increase, while the minority class error initially increases only decreasing as the number of iterations grow. The theoretical analysis carried out confirms that the gradient direction of the majority class dominates the training process leading to the slow convergence phenomenon that was previously observed by Anand et al. (1993). Moreover, the initial increase of the minority class error and its subsequent decrease is associated to changes in the gradient direction. In the initial iterations the angle between the global gradient and the majority class gradient is small leading to a fast decrease of the majority class error. However, the angle between the global gradient and the minority class gradient is above 90 degrees which results in an increase of the minority class error. After several iterations, the minority class error begins to decrease due to: (i) the majority class error becoming sufficiently small, and (ii) the angle between the global gradient and the minority class gradient being less than 90 degrees. Besides the theoretical justification, the authors experimentally tested multiple activation functions (Sigmoid and Tanh), cost functions (quadratic and cross entropy) and gradient descent algorithms (Batch Gradient Descent (BGD) and Momentum Stochastic Gradient Descent (MSGD)), showing that no combination of these alternatives is able to effectively reduce the negative impact of imbalanced data on both MLPs and CNNs.

The effects of class imbalance were also considered in the context of big data by Johnson and Khoshgoftaar (2020). The performance of a 2-layer and a 4-layer MLP was observed on two medicare datasets (Herland et al., 2018) representing highly imbalanced binary classification tasks. In both datasets, it was found that passing from 2 to 4 layers had a negative impact in the performance. This is the first work that investigates how changing

the number of layers affects the performance of MLPs in the context of big data. However, the exploration was still limited to two options for the number of layers on two datasets.

Valova et al. (2020) addressed another aspect of training CNNs under a multi-class imbalanced scenario by testing the performance impact of using different optimizers. The study considers 7 alternative optimizers, the Adam (Kingma & Ba, 2014), Rectified Adam (RAdam) (Liu et al., 2019), Yogi (Zaheer et al., 2018), and AdaBound (Luo et al., 2019), and also the combination of Adam with 3 different cyclical learning policies (Smith, 2017): the triangular Policy, triangular 2 Policy, and Exponential Range Policy, which are evaluated on the architecture heritage elements dataset (Llamas et al., 2017). In this dataset, there are two majority classes (gargoyle and column) while the remaining classes are less represented. The Yogi optimizer was found to provide the best training and testing results on different performance measures while Adam with Exponential Range Policy provides the second best results on the testing set. However, these results were obtained on a single dataset. The authors also tested Adam (Kingma & Ba, 2014), AdaGrad (Lydia & Francis, 2019), AdaDelta (Zeiler, 2012) and AdaBound (Luo et al., 2019) on the boats¹ dataset having reached the conclusion that Adam provides the best performance across all measures evaluated.

Bria et al. (2020) studied deep learning method on the specific imbalanced application of lesion detection in medical images. A CNN inspired by the VGGNet architecture (Simonyan & Zisserman, 2014) is used on the Microcalcification and Microaneurysm datasets. In this study the strong negative impact of imbalanced on CNNs was also confirmed. In effect, when learning the baseline CNN model, where the imbalance was the highest, the loss never decreased during the training although multiple attempts have been made using different training parameters.

2.3 Concept complexity and deep learning

Imbalance datasets pose an important challenge to deep learning as discussed in the previous section. However, it is also known that the complexity of the predictive task can influence the performance of traditional classifiers [e.g. López et al. (2013), Barella et al. (2021)]. In this section we review the relevant works that discuss the relation between concept complexity and deep learning.

Ho and Basu (2002) put forward several data complexity measures that can be clustered into 3 main groups: measures of overlap in feature values from different classes; measures of separability of classes; measures of geometry, topology, and density of manifolds. Table 2 shows a summary of the these groups of complexity measures.

A clear relation between these complexity metrics and the classifier performances was found by Cano (2013). While these metrics are mainly proximity-based, recent works point out to similarities between nearest neighbor and deep learning (Cohen et al., 2018). This allows us to argue that usage of such metrics may offer valuable insights into the performance of deep learning models. However, neither the class imbalance problem nor deep learning methods were taken into account in this study. Santos et al. (2022) presented recently an extensive study on the joint-effect of class imbalance and overlap. This work provides a very interesting taxonomy of class overlap measures.

¹ <https://www.kaggle.com/kunalgupta2616/boat-types-recognition>.

Table 2 Complexity measures defined by Ho and Basu (2002)

Category	Measure	Description
Measures of overlaps in feature values from different classes	F1	Maximum Fisher's discriminant ratio
	F2	Error rate of linear classifier by linear programming
	F3	Maximum (individual) feature efficiency
	L1	Minimized sum of error distance by linear programming
Measures of separability of classes	L2	Error rate of linear classifier by linear programming
	N1	Fraction of points on class boundary
	N2	Ratio of average intra/inter class NN distance
	N3	Error rate of 1NN classifier
Measures of geometry, topology and density of manifolds	L3	Nonlinearity of linear classifier by linear programming
	N4	Non-linearity of 1NN classifier
	T1	Fraction of points with associated adherence subsets retained
	T2	Average number of points per dimension

Recently, Barella et al. (2021) showed that the original data complexity measures are not suitable for an imbalanced scenario with real-world data, and thus the authors discouraged their use. As an alternative solution the authors provided simple adaptations of the original metrics which were found useful for determining the difficulty faced by classification algorithms when dealing with an imbalanced setting. Although this is an interesting contribution, these adaptations were only developed and assessed for binary classification problems.

Luengo et al. (2011) took the analysis of the data complexity measures to another level in the class imbalance context by using them to evaluate the behavior of different pre-processing methods. Rules describing a good or bad behaviour of the pre-processing methods are built using measures F1, N4 and L3. However, this study was limited to C4.5 and PART algorithms and to 3 measures.

Other works (e.g. López et al. (2013); Dudjak and Martinović (2021)) studied data intrinsic characteristics that hinder the learning process when the data is imbalanced. López et al. (2013) analysed 6 characteristics, namely: the presence of small disjuncts, the lack of density in the training data, the class overlap, the presence of noisy instances, the significance of the borderline examples, and the distribution shift between the training and test data. These data intrinsic characteristics are found to pose a strong handicap when learning from imbalanced datasets. Although useful and relevant, this analysis was carried out with the C4.5 classifier and no deep learning model was evaluated. Dudjak and Martinović (2021) ranked these difficulty factors in terms of their negative impact on the learning performance. The results suggested that the class imbalance problem impacts the classifiers performance when this problem is combined with other data intrinsic characteristic. On the other hand, the negative effects observed for datasets containing these intrinsic characteristics are more visible when the class imbalance is also present. The authors carry out an analysis to observe which classifiers are can cope well with class separation into sub-concepts. Dudjak and Martinović (2021) state that the MLP is one of the algorithms that cannot conceptually handle the presence of small disjuncts. For the classifiers that can deal with this, noise is the characteristics with the most detrimental effect on the performance, followed by class overlap and class imbalance.

The interest in studying the class imbalance problem has grown significantly since the rise of deep learning. However, only a few works consider the issue of task complexity in this setting. Among these works, several use an intuitive notion of the difficulty of the tasks. For instance, Buda et al. (2018) stated that the three benchmark datasets considered in the study, together with the corresponding CNN models selected for each one, are of increasing complexity. However, no specific complexity measure was calculated. Johnson and Khoshgoftaar (2020) presented datasets as having different complexity, although no measure for this complexity was provided. The problem of insufficient class samples was also considered in this study being artificially achieved by varying the class size. The results revealed a different behaviour for each dataset considered, showing a particular sensitivity to the class imbalance and sample size issues.

Murphrey et al. (2004) proposed three methods for measuring the noise level in a dataset using its characteristics. The first method measures the noise level through intra- and inter-class distances; the second method calculates the linear separability of the dataset; and the last method measures the number of examples from the opposite class that fall within the bounding box of each class. Neural networks performance was shown to greatly depend on the classes separability. For high levels of noise, none of the tested networks (MLP, ART and RBF) performed well in an imbalanced scenario.

Raj et al. (2016) proposed a class separability score based on the Silhouette metric and showed that both the imbalance ratio and the class separability have a negative impact on the performance of multi-channel CNNs in the context of time series data.

Finally, Sáez et al. (2016) proposed to analyze the complexity of each class in multi-class imbalanced setting by taking into account the instance-level difficulties. An analysis of homogeneity of the neighborhood of minority class instances lead to definition of four types of minority instances. Their frequency in each minority class allowed to rank them according to their complexity and use this information to improve oversampling. This concept was further extended by Sleeman and Krawczyk (2021) onto massive imbalanced data sets analyzed on high-performance clusters.

Overall, we observe that the research in this field is still limited, requiring more extensive experiments. We observe that frequently an intuitive notion of complexity is used. Moreover, there are studies that look into other data characteristics that may cause an additional difficulty to the class imbalance problem. This is also an interesting perspective that is still not sufficiently explored in the context of deep learning. More extensive experiments and the development of ways to determine the complexity of tasks are still necessary.

3 Experiments set up

Motivation The literature review of the previous section suggests that deep learning systems are not immune to the class imbalance problem. Furthermore, two papers provided insights as to the reason for this by observing the size and direction of the gradients associated with the majority and minority classes. While all the work surveyed concluded that class imbalances are harmful, and many looked at various factors including increasing class imbalances, increasing the number of minority classes, increasing network depths, using different optimizers, and considering class imbalances along with concept complexity, none of them looked systematically at the relationship between class imbalance, concept complexity and scarcity of the data, three factors known to be important in the traditional learning (e.g., symbolic, statistical) or shallow neural-network context. The purpose of this part of the paper is to establish this correspondence in the simplest context, that of binary classification, as a first line of inquiry. In addition, the work discussed in the previous section did not establish the relation between network depth and class imbalances as different papers made different observation on this topic: Ding et al. (2017) found that depth in CNNs helped a bit (e.g., a 6-layer CNN did not perform as well as a 10-layer one), but that beyond a certain point (past 10 layers), depth was helpful in speeding up convergence, but not obtaining better performance. Johnson and Khoshgoftaar (2020), on the other hand, found in its experiments that depth hurt the performance of an MLP network. Establishing a relationship between class imbalances, complexity, size and network depth is also part of our quest.

Experimental goals In more detail, the controlled experiments we set up aim at answering three research questions:

RQ1: In traditional learning systems, there is a tight dependency between class imbalances, concept complexities, dataset size and the performance of the classifier. Our literature review suggests that deep learning systems also suffer from the class imbalance problem, but does it do so for the same reasons as traditional learning systems or does

deep learning alleviate some of the factors (e.g., concept complexities) at the root of the problem?

RQ2: What specific role, if any, does the depth of the network play in the imbalance/complexity/scarcity triangle?

RQ3: Does regularization modify that role?

We answer these questions in the context of two different types of deep learning systems: Multi-Layer Perceptrons (MLP) and Convolutional Neuronal Networks (CNN) and two different kinds of domains: Artificially generated and Real².

Experiments with concept complexity The literature review of the previous section also indicates that only few studies were proposed that combine observations regarding the complexity of the concept, the class imbalance problem and deep learning. Furthermore, the notion of concept complexity used in these studies was usually rather intuitive than formal. In this work, we directly address the relationship between concept complexity, class imbalance and deep learning, and we handle the question of measuring concept complexity in two ways. First, we use artificial domains that naturally present problems of increasing complexity. In the backbone domain used in the MLP experiments, complexity is represented by the number of separating hyperplanes needed to correctly classify the domain, while in the Shapes domain, used in the CNN experiments, the difference in the type of shapes used in both classes account for the complexity of the task. Second, we developed a procedure to assess domain complexity in the real world domains. Our method consists of three steps: (1) T-SNE projection of the domain; (2) Visual pre-selection of candidates for different degrees of complexity; (3) Selection through rigorous cross-validation experiments on a subset of the domain.

Experiments with data scarcity Another line of enquiry of our experiments concerns the overall size of the training set. In the artificial domains, we experiment with a small size where the amount of data available is too scarce; and a large size where the data is sufficiently represented. We did not experiment with this factor in the real domains since data scarcity is the default condition.

3.1 Deep multilayer perceptrons

We start by considering the case of MLP networks. We first describe the domains that were historically used by the community to answer our questions. We then describe our experimental set-up.

3.1.1 Artificial domains

Preliminaries To create a family of domains appropriate to establish the relationship between predictive performance, class imbalance, domain complexity and data scarcity, we followed the approach proposed in Japkowicz and Stephen (2002) to generate domains that vary according to three dimensions: overall size of the data set (s), structural concept complexity (c), and degree of balance between the classes (b). The family of domains created by this approach was shown to reflect some of the main challenges surrounding the class

² The code for the experiments can be accessed here: <https://github.com/cbellinger27/deep-imbalance-analysis>.

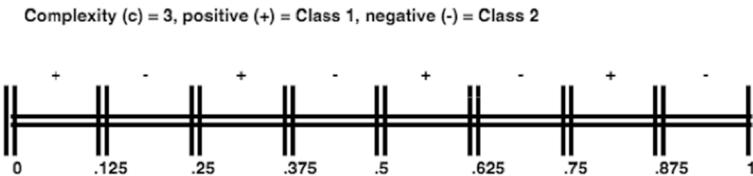


Fig. 1 Domain backbone of Complexity 3. In this one-dimensional family of domains, the complexity of the task increases as the number of alternating sub-concepts of each class increases

imbalance problem and was, therefore, deemed relevant to apply in the case of the deep learning approaches under consideration in this work.

Domain generation Fifty domains were generated as follows: each domain is uni-dimensional with inputs in the $[0, 1]$ range associated with class 1 (+) or 0 (−). The $[0, 1]$ input range is divided into sub-intervals of the same size, each associated with class value 0 or 1. Contiguous intervals have opposite class values. The complexity level, c , can take values from 1 to 5. Depending on its value, different numbers of sub-intervals are created. An example of a backbone model is shown in Fig. 1. The backbone in the figure can be understood as representing a two-class domain where each class is composed of 4 subconcepts that are mixed amongst themselves. Hypothetically, for example, the reader can imagine that the data located between 0 and .125 represents the subspecies of dogs A which is very closely related to the subspecies of wolves A located between .125 and .25, while the location between .25 and .375 represents subspecies of dogs B, while the location between .375 and .5 represents subspecies of wolves B; and so on.

Structural concept complexity The value of c is used to determine the number of sub-clusters present in the backbone that ranges within $[0, 1]^3$. The number of subclusters is calculated as 2^c and the width of each of these sub-sections is calculated as $\frac{1}{2^c}$. As illustrated in Fig. 1, the distribution of Class 1 (+) and Class 0 (−) is determined by assigning them regular, alternating sub-intervals. This is done regardless of the size of the training set or its degree of imbalance. Once the backbone is generated based on the value of c , actual data points are generated within each sub-interval by generating points at random using a uniform distribution. The number of points sampled from each interval depends on the size of the domain as well as on its degree of imbalance.

Dataset size Our investigation revolves around two dataset sizes, which we will refer to as sizes 1 and 5 (or $s=1$ and $s=5$), according to Japkowicz and Stephen (2002). In Japkowicz and Stephen (2002), prior to considering the balance level, b , the total number of examples in the size 1 experiments is calculated as $\left(\frac{5000}{32} \times 2\right)$ where each sub-interval contains $\left(\frac{\frac{5000}{32} \times 2}{2^c}\right)$ examples. In the size 5 experiment, the dataset holds a total number of $\left(\frac{5000}{32} \times 2^5\right)$ examples with $\left(\frac{\frac{5000}{32} \times 2^5}{2^c}\right)$ instances in each of the sub-intervals.

³ It is well known from the traditional machine learning literature Prati et al. (2004) that class overlap is also a major factor in the study of class imbalances. We considered it in Ghosh et al. (2021), but after observing that its effect was minimal in the case of deep learning, we decided not to include it in this article. The effect is naturally present in the real world domains considered, so it has been, implicitly considered here.

Table 3 Training and Testing Balance levels used in all the artificial datasets (Backbone and Shapes)

Large size	Number of training instances		Small size	Number of training instances		Large/ small size	Number of testing instances	
	Class 1	Class 2		Class 1	Class 2		Class 1	Class 2
B1	5000	128	B1	156	10	B1	5000	5000
B2	5000	263	B2	156	20	B2	5000	5000
B3	5000	882	B3	156	39	B3	5000	5000
B4	5000	2143	B4	156	78	B4	5000	5000
B5	5000	5000	B5	156	156	B5	5000	5000

Once the basic number of instances per sub-interval is determined, we decrease that number for class 0, the minority class, according to the degree of balance, b . Meanwhile, the number of instances in the Class 1 sub-intervals representing the majority class remain the same as discussed in the previous paragraph. The number of instances belonging to the Class 0 sub-intervals is calculated as $\left(\frac{\left(\frac{5000}{32} \times 2 \right)}{2^b} \right)$ for size 1 and $\left(\frac{\left(\frac{5000}{32} \times 5 \right)}{2^b} \right)$ for size 5. The expression $\left(\frac{32}{2^b} \right)$ gives a limit of 5 to the degree of balance in our experiment. When $b=5$, the number of instances in each of the sub-intervals is the same and the data set is perfectly balanced. This states that the value of b is inversely proportional to the disparity or the degree of imbalance between the classes.

Table 3 shows the result of these calculations for both the large ($s = 5$) and small ($s = 1$) data sets. These are the training numbers used for all the experiments on artificial data sets (Backbone and Shapes). The testing numbers are constant for all the results as per our balanced testing approach as discussed below. In the real world domains in the Text and Image Domains, we used a different formula to compute the number of instances. The numbers used in these experiments will be reported with the description of the real world domains.

3.1.2 Real domains

Preliminaries To incorporate realistic data in our experiments, we considered two text domains to understand how the findings from the artificial domains translate to real-life text domains. This is beneficial in understanding whether supplementary steps are required to attain reliability in the performance. For these experiments we utilised two different datasets: 20NewsGroup (Alhenaki & Hosny, 2019) and Job Classification.⁴ Special characters, punctuation, and stopwords are removed from each dataset instance. Stemming and lemmatization were used to further pre-process the data, as well as the replacement of acronyms with their full forms.

Complexity-driven dataset selection Binary problems were selected using a combination of visual inspection of binary T-SNE plots and cross-validation experiments. The aim of the selection was to identify five binary domains with increasing levels of complexity:

⁴ <https://www.kaggle.com/adarshsng/predicting-job-type-category-by-job-description?select=train.csv>.

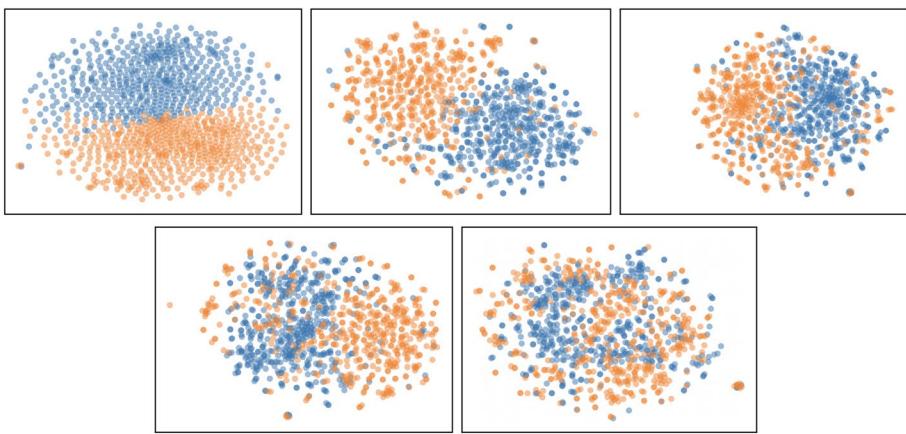


Fig. 2 T-SNE Plots of Binary 20Newsgroup datasets sorted from the least to the most complex

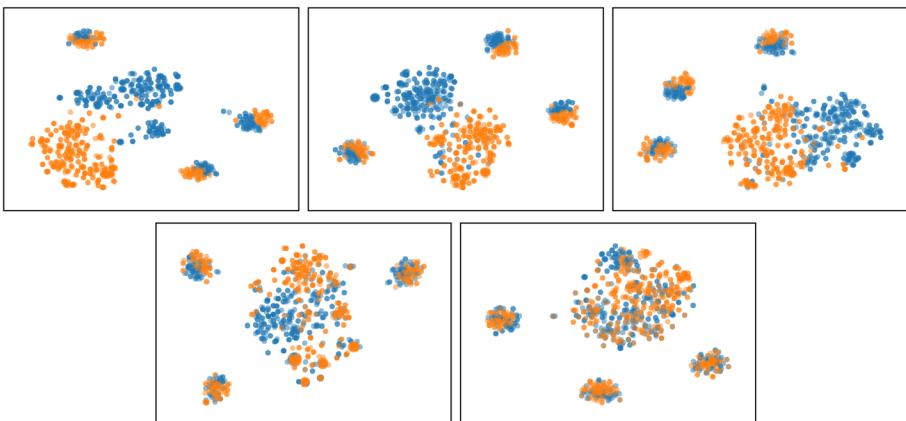


Fig. 3 T-SNE Plots of Binary Job Classification datasets sorted from the least to the most complex

starting from an easy domain where points appear mostly linearly separable, to a moderate domain characterized by the structural concept complexity and overlapping phenomena, to more extreme and difficult scenarios that suffer from both issues simultaneously.

In order to select the domains, we sampled 500 posts from each class and exhaustively generated plots for all pairwise combinations of classes. After visually pre-selecting the most relevant domains in each category, we performed experiments to confirm the complexity of the selected scenarios and ranked them accordingly. In particular, we adopted the average G-Mean performance achieved using 2x10-fold stratified cross-validation⁵ on the binary settings with different model architectures (from 1 up to 5 hidden layers) using balanced data. The selected domains ordered by increasing level of complexity are shown in Figs. 2 and 3.

⁵ 2x10-fold stratified cross-validation is used to ensure stable means and standard deviations in the results.

Table 4 Number of training instances in the MNIST fashion and CIFAR domains (left) and 20 Newsgroup (right)

	CIFAR/MNIST	Number of Training Instances		20 News Group	Number of Training Instances	
		Class 1	Class 2		Class 1	Class 2
B1	5000	125	B1	565	14	
B2	5000	250	B2	565	30	
B3	5000	750	B3	565	100	
B4	5000	1500	B4	565	242	
B5	5000	5000	B5	565	565	

3.1.3 Experimental set-up

Train/test regimen We tested our networks using the balanced testing approach by forming a testing dataset containing 1000 examples in each of the sub-clusters of the proposed backbone, i.e., 1000 instances per class for $c = 1$; 2000 per class for $c = 2$, etc., up to 5000 per class for $c = 5$. This not only provides an unbiased testing framework, but also provides a wide variety of data to test on. This approach, therefore, helps understand the actual potential of the classification models. We report the Balanced Testing results in the paper.

Although we calculated our results based on a variety of metrics, we decided to report the G-Mean after noticing that other metrics such as the F1-measure and the balanced average all lead to the same conclusions. These results are available upon request. We, again, used the balanced testing approach with the same number of minority and majority class used for testing.

Table 4 (right) shows the number of training instances used in the 20 Newsgroup data sets. The testing numbers are constant for all the results as per our balanced testing approach, as discussed previously, and were set to 565 per class in the 20 Newsgroup domain. Given the constraints of the real world domains, we didn't use the formulae, for size, defined for the artificial domains. For the Job Classification domain, we could not use the same number of instances at each complexity levels as we did in the 20 Newsgroup domains since, in order to obtain problems of different concept complexity, we needed to use different pairs of categories and each category had different numbers of instances. As a result, we couldn't summarize the number of instances used in that domain in a single table. Instead, we report these numbers in Fig. 17 in the appendix.

Models and their parameters The experiments are conducted on five different depth of MLP to show how the linear effect of increasing the depth of MLPs affects classification. Each of the MLP models are termed Model-x where x stands for the number of hidden layers and takes a value between 1 and 5. We start from the shallowest model (Model-1) and reach up to the deepest model with 5 hidden layers (Model-5). For each of the networks, we report the optimal results recorded after experimenting with 2, 4, 8, and 16 Hidden Units (HU) in each layer. We trained each of the MLP networks for 300 epochs, with a learning rate of 0.001, using the Adam optimizer. Relu activation and uniform weight initialization were utilized as recommended in Glorot and Bengio (2010) to reduced the risk of vanishing gradients. We also experimented with three different types of regularization approaches: Dropout, Reduce on Plateau and Early stopping.

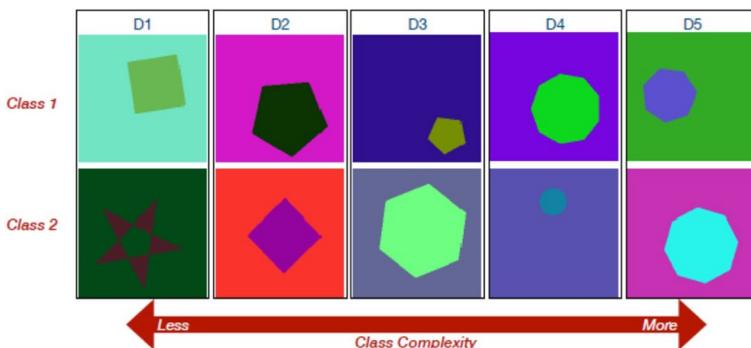


Fig. 4 The five degrees of complexity in the Shape Domain

3.2 Convolutional neural networks

We now consider the case of CNN networks. Once again, we first describe an artificial domain that was borrowed from the community to answer our questions and we then describe our experimental set up.

3.2.1 Artificial domains

Dataset characteristics For our experiments with CNNs, we used a data set proposed by El Korchi and Ghanou in 2020. The dataset consists of multi-coloured images of shapes with the following characteristics:

- Each image is of size $200 \times 200 \times 3$
- In each image, shapes and backgrounds have solid colours
- In each image, shapes and background colours are random
- In each image, shapes sizes and rotations are random

Figure 15 in the appendix shows the kind of variability found within a class, in this case the Square class. Taking this data set as a point of departure, we created five basic classification problems with an increasing level of complexity. In this case, the notion of complexity is different from the notion of complexity defined by the backbone model. Specifically, the complexity is based on the visual similarity between the shapes (caused by the number of angles separating two classes in the polygon category of shapes). This is illustrated in Fig. 4 where the problem of squares versus stars is the simplest problem, followed by the pentagon versus square, hexagon versus pentagon, circle versus nanogon and octagon versus heptagon problems, in order of difficulty. As a result, the notion of complexity is related to the size of the margin separating the two concepts.

Dataset modifications We modified the original data sets by reducing the size of images to $32 \times 32 \times 3$ in order to make our experiments more computationally tractable and we added noise to make the data more realistic. Figure 16 in the appendix illustrates the five classification problems in modified mode. We artificially sampled the data set to create five different balance levels, b. The testing set consists of a balanced data set no matter what the

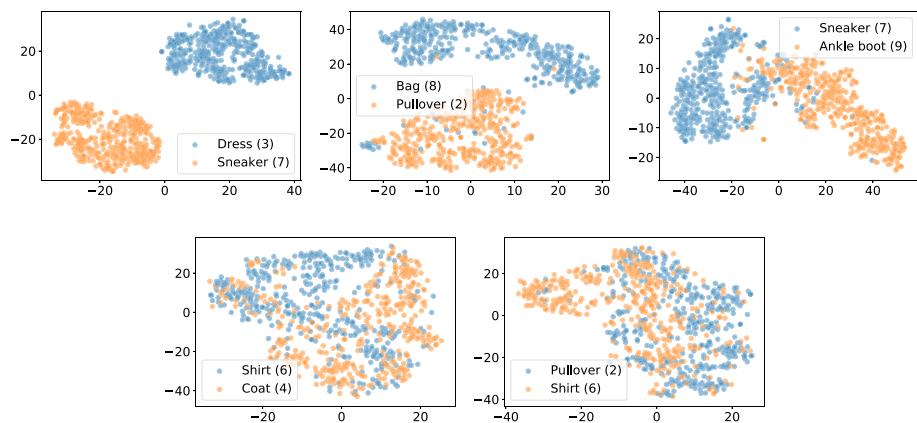


Fig. 5 T-SNE Plots of Binary MNIST Fashion datasets sorted from the least to the most complex

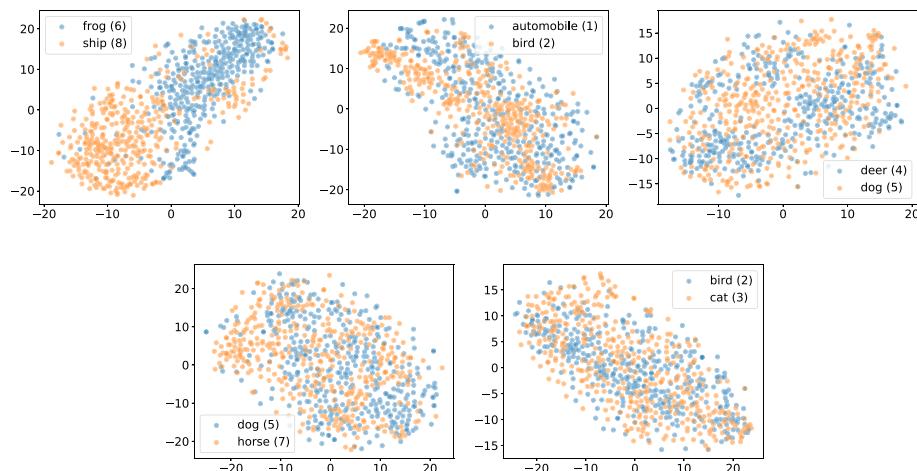


Fig. 6 T-SNE Plots of Binary CIFAR10 datasets sorted from the least to the most complex

imbalance level of the data was. The balance levels we created are the same as those used in the Backbone domain and were shown in Table 3.

3.2.2 Real domains

We considered two image domains to see how the conclusions drawn from the artificial domains' experiments translate to realistic domains and whether additional considerations

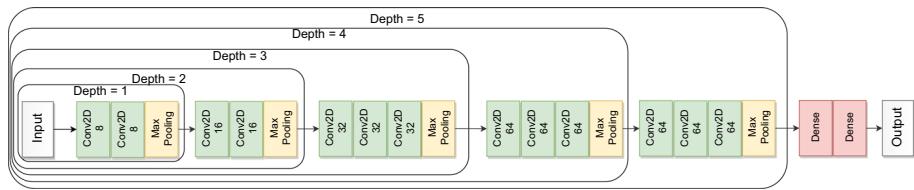


Fig. 7 CNN model architecture adopted in experiments with Image datasets, with increasing levels of depth. The numbers mentioned with the convolutional layers (Conv2D) denote the number of filters used in each layer

need to be taken. For these experiments, two benchmark datasets are considered: Fashion-MNIST⁶ and CIFAR10⁷.

Once again, we considered 5 balance levels, $b = \{5, 4, 3, 2, 1\}$, where $b = 5$ represents the fully balanced level and $b = 1$, the most imbalanced one. The exact number of instances per class is given in Table 4 and, as for the other domains used in this study, the minority classes of all the unbalanced domains were created by randomly undersampling the second class (considered the minority class) of each originally balanced binary domain.

Domains As in the case of the Text classification domains and using the exact same procedure, binary domains were selected using a combination of visual inspection of binary T-SNE plots and cross-validation experiments. The selected domains ordered by increasing level of complexity are shown in Figs. 5 and 6.

3.2.3 Experimental set-up

Train/test regimen We tested the CNNs in the same way as we tested the MLPs, using the balanced testing approach. We formed a testing dataset containing 5000 examples of each class. As previously discussed, this not only provides an unbiased testing framework, but also provides a wide variety of data to test on. This approach, therefore, helps understand the actual potential of the classification models. We report the Balanced Testing results in the paper. The stratified cross-validated results are available upon request.

As for the MLP experiments, we calculated our results based on a variety of metrics and decided to report the G-Mean after noticing that other metrics such as the F1-measure and the balanced average all lead to the same conclusions. These results are available upon request.

Table 4 (left) shows the number of training instances used in the MNIST-Fashion and CIFAR-10 data sets. The testing numbers are constant for all the results as per our balanced testing approach, as discussed previously, and were set to 5000 per class for both domains. As before, given the constraints of the real world domains, we didn't use the formulae, for size, defined for the artificial domains.

Models and their parameters The model architecture considered in this set of experiments is a CNN with an increasing number of convolutional layers (filters): 1 (8), 2 (8-16), 3 (8-16-32), 4 (8-16-32-64), 5 (8-16-32-64-64). Two dense layers are featured at the end of each model architecture. As in the previous experiments, relu activation and uniform

⁶ <https://github.com/zalandoresearch/fashion-mnist>.

⁷ <https://www.cs.toronto.edu/~kriz/cifar.html>.

weight initialization were utilized in these models (Glorot & Bengio, 2010). A graphical representation of the architecture considered is shown in Fig. 7.⁸

We also experimented with three different types of regularization approaches: Dropout, Reduce on Plateau and Early stopping.

4 Analytical framework

As previously discussed, this paper has two goals: first, to review the literature devoted to class imbalances and deep learning and second, to answer the three research questions (RQ1, RQ2 and RQ3) presented at the beginning of Section 3. In this part of the paper, we are concerned with the second goal.

Structure of the answers We now reiterate the three research questions and explain how we set out to answer them through our experiments.

□ *RQ1 asks: In traditional learning systems, there is a tight dependency between class imbalances, concept complexities, dataset size and the performance of the classifier. Our literature review suggests that deep learning systems also suffer from the class imbalance problem, but does it do so for the same reasons as traditional learning systems or does deep learning alleviate some of the factors (e.g., concept complexities) at the root of the problem?*

We structure our answer in the context of the findings that were made in the traditional learning case. In more detail, Japkowicz and Stephen (2002) makes four important observations regarding the dependency between class imbalance, sample size and concept complexity. It does so mostly in the context of decision trees, but expands the analysis to shallow MLPs and SVMs as well. The main observations reported in Japkowicz and Stephen (2002) study are:

1. Linearly separable domains are not sensitive to class imbalance.
2. In non-linearly separable domains, as the class imbalance increases, so does the amount of misclassification.
3. The class imbalance problem is exacerbated by the problem's complexity.
4. The class imbalance problem is exacerbated by data scarcity.

In our result analysis, we will assess which of these observations still hold in the deep learning context for both MLP and CNN; and whether the observations hold in both artificial and natural data sets. Section 6 will discuss the case of MLP, while Sect. 7 will move to the case of CNN. Once RQ1 is answered, we move on to RQ2.

□ *RQ2 asks What role does the depth of the network play in the imbalance/complexity/scarcity triangle?* We answer this question by adding depth considerations to our previous observations. In particular, we show the results obtained at all depths considered in order to gauge its effect.

□ *RQ3 asks Does regularization modify that role?* We answer the question by adding three regularization techniques: Dropout, Reduce on Plateau and Early stopping and showing which combinations improve the results most and by how much.

⁸ We consider that the depth of the CNN is represented by its total number of trainable layers as shown in Fig. 7.

Tools used in our answers We answer the questions with the help of barplots and heatmaps. The barplots show the G-Means obtained by the deep networks under scrutiny on the different domains. The first sets of barplots in each section display the results in terms of class imbalances, concept complexity and data set size. These are used to answer RQ1. The second set of barplots throws the additional question of network depth into the equation in order to answer RQ2. For readability, the body of the paper shows only one such set of barplots. The other three are shown in the appendix for each of MLP and CNNs (i.e., 6 additional sets of barplots altogether). The subsequent heatmaps summarize the optimal network configuration, in terms of depth, and shows which regularization techniques improved the results further, as per RQ3. A discussion based on resulting barplots displayed in the appendix discusses the size of the improvement in each case.

Organization of our answers Section 8 summarizes the main takeaways of our experiments without showing the actual results or discussing any detail in depth. We thought it prudent to provide such a section, given that we conducted many experiments and obtained a number of intricate results, which we felt could detract from the main issues which we believe can help the field move forward. Sections 6 and 7 dive into the details left out from Sect. 8 by showing the results obtained by our experiments and conducting their analyses in the context of MLP, in the case of Sect. 6 and CNN, in the case of Sect. 7.

5 Notational details

Prior to turning to the more detailed results, we remind the reader about our notations. The reader can turn back to this description when reading the results of the next two sections:

- b, the balance level, is used to designate the level of balance in the data set. $b = 5$ corresponds to perfect balance, while $b = 1$ corresponds to the highest degree of imbalance.
- c, the degree of complexity, is used to indicate how complex a domain is. $c = 1$ represents the simplest domain while $c = 5$ represents the most complex domain of a series of domains.
- s, the sample size, indicates whether we are using a large domain for our experiments ($s = 5$) or a small one, in which the data is rather scarce ($s = 1$).
- d, the depth of the network, corresponds to the number of layers used in the system. $d = 1$ corresponds to the shallowest setting, while $d = 5$ corresponds to the deepest.

6 Results in the multi layer perceptron case

Overview We now discuss the results obtained with MLP in detail. We first discuss the dependency between class imbalances, concept complexity and sample size; we then test whether the depth of the networks used makes a difference; and we conclude with an analysis of the role of regularization on the results.

Summary of main observations Conducted experiments showed that deep MLP models are as much affected by class imbalance than their shallow counterparts. Furthermore, for a number of analyzed learning difficulties we can observe that increasing the depth of these models is beneficial to their robustness and predictive power. Finally, we have not observed any significant gains from employing regularization approaches under class imbalance.

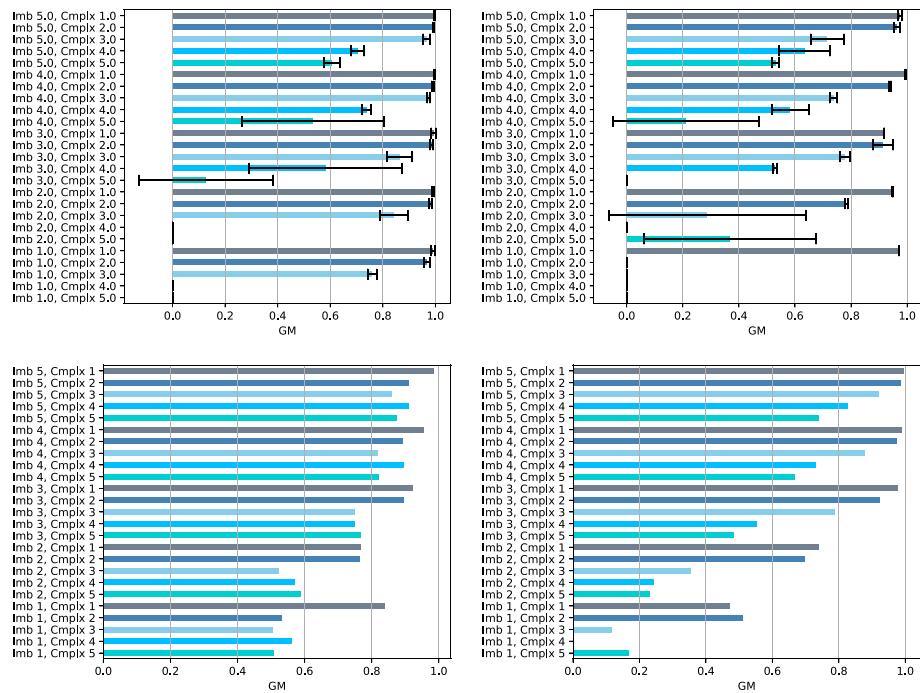


Fig. 8 G-Mean results of applying deep MLP models under optimal depth conditions. Top Left: Large Backbone problem ($s=5$); Top Right: Small Backbone problem ($s=1$). Bottom Left: 20 Newsgroup problem. Bottom Right: Job application problem. (Low complexity: level 1; High complexity: level 5. High imbalance: level 1; Low imbalance: level 5.)

6.1 Answering RQ1: what is the dependency between class imbalance, concept complexity, sample size and deep learning?

The results obtained by MLP networks are shown in Fig. 8 for the backbone models and the text domains. We discuss to what extent each of the four observations made in Japkowicz and Stephen (2002) translate to the deep MLP setting. It is important to note at the outset that Japkowicz and Stephen (2002) had reported that while shallow MLP networks ($d = 1$) showed the same trends as C5 decision trees, the MLP results showed more variance and, generally speaking, MLPs suffered from the class imbalance problem less than Decision Trees. We consider each of the observations from Japkowicz and Stephen (2002) in turn and pit them against the graphs of Fig. 8.

⇒ *Observation 1 In traditional learning approaches, linearly separable domains are not sensitive to class imbalance*

Backbone domains In deep MLPs, we observe the same trend on the backbone domains for both data set sizes. A very slight decrease in G-Mean can be observed as the class imbalance increases, but it is very slight and is comparable to what happened in the original study.

Text domains It is not clear that this question can be clearly considered in the text domains since as shown in the T-SNE plots of Figs. 2 and 3, the least complex domains

do not appear to be linearly separable in the 2-dimensional projections. In the results obtained on both the 20 Newsgroup and Job Description problems, shown in the bottom row of Fig. 8, we see that as the balance level decreases so does the performance on the simplest domain of complexity $c = 1$. While the drop in performance is constant as the balance decreases, a large negative step is seen in both domains at balance level $b = 2$.

⇒ *Observation 2 In traditional learning approaches, as the class imbalance increases, so does the amount of misclassification*

Backbone domains In deep MLPs, on the backbone domains, the same trend is observed, starting at balance level $b = 3$ down to $b = 1$, for large data sets, and $b = 4$ down to $b = 1$ for small data sets.

Text domains The same trend is observed in the text domains. While there is a continuous decrease in performance with respect to the increase in class imbalance, the impact of class imbalances becomes generally significant (for all concept complexities) at balance level $b = 3$ for the 20 Newsgroup problem and $b = 2$ for the Job Classification problem.

⇒ *Observation 3 In traditional learning approaches, the class imbalance problem is exacerbated by the problem's complexity*

Backbone domains In deep MLPs, the same trend is observed in both the large and smaller data sets. In the larger data set ($s = 5$), this trend is seen particularly well for problem complexity $c = 5$, starting at balance level $b = 3$ down to $b = 1$, but even for complexities $c = 3$ and 4 , at balance levels $b = 2$ and $b = 1$ for the large data set. In the small data set, the exacerbation of the problem is seen as early as for balance level $b = 4$ for complexity $c = 5$. It is very clear at class balances $b = 1$ and 2 for complexities $c = 2, 3, 4$ and 5 .

Text domains This trend is also observed in both text domains, as shown in the bottom left graphs of Fig. 8. For example, following two sets of bars, say the dark blue bar representing concept complexity 2 and the green bar representing concept complexity 5 at all degrees of imbalance, we see that the drop in performance as the class balance decreases is larger for the green bar than it is for the dark blue one, except from balance levels $b = 2$ to $b = 1$, where the problem has become so difficult for the most complex domain, that there is less change in performance, whereas the simpler concept still has “room” to drop performance.

⇒ *Observation 4 In traditional learning approaches, the class imbalance problem is exacerbated by data scarcity⁹*

Backbone domains In deep MLPs, the trend is a bit different as the results degrade due to dataset size starting at class balance $b = 4$ and concept complexity $c = 2$.

Text domains This trend was not tested in the text domain as we did not have access to a large data set.

□ *RQ1 for MLP answered* All in all, this analysis concludes that deep MLP behave pretty much in the same way as their traditional learning counterparts, with regard to the class imbalance problem.

⁹ This was particularly true in the case of Decision trees though it was generally not the case for shallow MLPs.

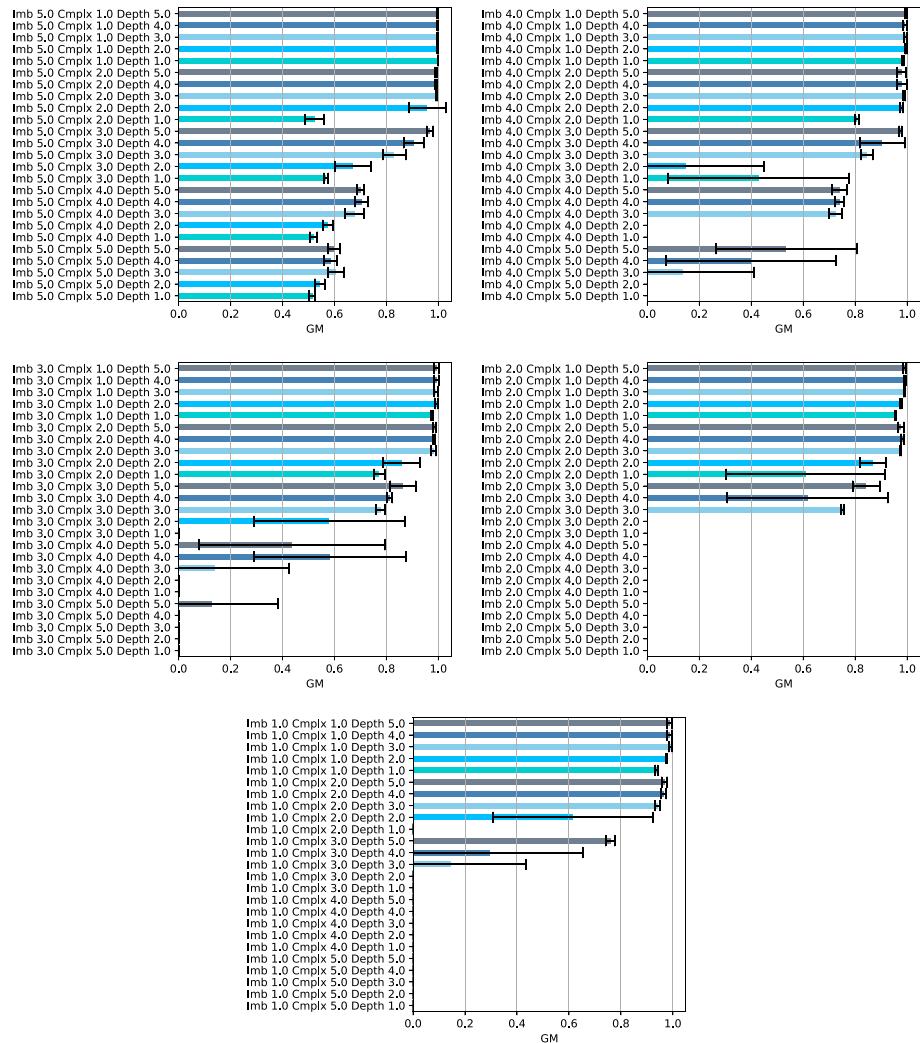


Fig. 9 The effect of depth on class imbalance levels, complexity in large datasets in the Backbone domain. Plots by increasing imbalance level (leftmost: least imbalanced; rightmost: most imbalanced). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

6.2 Answering RQ2: how does the depth of the networks affect the imbalance/complexity/performance triangle?

In the previous section, we explored the effect of domain characteristics on the performance of DL models without taking their depth into consideration. Indeed, the results we reported were obtained at different depths since our experiments consisted of training networks of different depths (1 to 5) on all domains and selecting the best results for each domain, a-posteriori. In this section, we explore whether there is any evidence, from the optimal depths selected in these experiments, that could allow us to link the depth of a

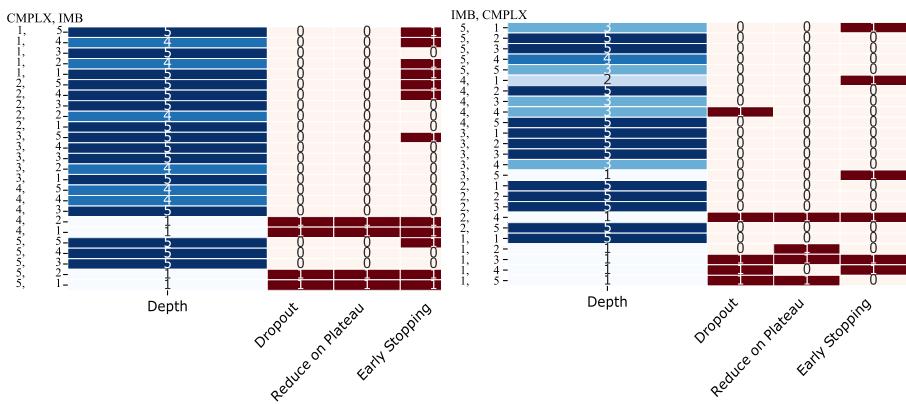


Fig. 10 Optimal Depth and Regularization settings for MLP experiments for large data set size ($s = 5$, left) and small dataset size ($s = 1$, right)

network to its aptitude at dealing with class imbalances or other complex domain characteristics. This was achieved by considering all the depths considered in each setting.

Backbone domains From Fig. 9, it is clear that depth plays a role in the question of concept complexity even when considering a large dataset size and no class imbalances. Indeed, in the top left plot of Fig. 9 ($s = 5$ and $b = 5$), we can see that at concept complexity $c = 1$, all network depths obtain the same perfect performance, at concept complexity $c = 2$, depths $d = 5, 4$, and 3 obtain the same perfect performance, depth $d = 2$ obtains a slightly lower performance and depth $d = 1$ obtains a significantly lower one. At concept complexity $c = 3$, going from depth $d = 5$ to depth $d = 1$ causes a linear decrease in performance. A decrease in performance is still noticeable at concept complexity $c = 4$, although all the performances are lower and the differences are smaller. Finally for concept complexity 5 , the depth does not seem to play much of a role as the performance is low, hovering around .5 and .6. Considering the equivalent situation of no class imbalances in the small dataset size setting—the top left plot of Fig. 18 in the appendix—we notice a similar pattern for complexity level $c = 1$, good performance for depths $d = 5$ and 4 in the case of complexity $c = 2$ and a drop in performance for depth $d = 3$ (which is unstable, as suggested by the size of the standard deviation bars), depths 2 and 1 (which are stable but display low performance, just below .6). In all other cases, depth has little effect as the problems appear too difficult for the networks to handle satisfactorily.

Turning now to the question of class imbalances, we see that there again, the layer of difficulty added by the class imbalance problem show the advantage of adding depth to the networks. In the large dataset case (Fig. 9), considering concept complexity $c = 3$, for example, we see that at class balance level $b = 4$, Depth $d = 5, 4$ and 3 relate to performance approximately in the same way as it did for class balance level $b = 5$, but Depths $d = 2$ and 1 show a decrease in performance and stability not seen at class balance level 5 . That trend is even more pronounced at balance level $b = 3$, where depth $d = 1$ leads to a performance of 0, and further deterioration follows at class balance levels $b = 2$ and 1 . Similar and even more pronounced types of deterioration can be seen in the case of the small dataset (Fig. 18 in the appendix). The observations made for concept complexity $c = 3$ also apply to various degrees to different concept complexity levels.

Text domains The results obtained on the text domains are displayed in Figs. 19 and 20 in the appendix. We notice two effects that depth can have on performance in certain cases (in many other cases, depth does not play a major role on the performance of the multilayer perceptrons on the 20 Newsgroup and Job Classification problems).

The first noticeable effect is that there are situations where a low depth of $d = 1$ is detrimental to the performance of the network. This happens, most noticeably, at balance levels $b = 1, 2$, concept complexity $c = 1$ and depth $d = 1$ and balance level $b = 2$, concept complexity $c = 2$ and depth $d = 1$.

The second noticeable effect is that there are situations where a high depth of $d = 5$ is detrimental to the performance of the network. This happens, most noticeably, at balance levels $b = 5, 4, 3$ and 2 , concept complexity $c = 5$ and depth $d = 5$ and to a lesser extent at balance levels $b = 5$ and 4 , concept complexity $c = 3$ and depth $d = 5$.¹⁰

□ *RQ2 for MLP answered* Altogether, this suggests that increasing the depth of the MLP is helpful in difficult domains afflicted by concept complexity, data scarcity, and class imbalance. However, as shown in the figures, it is clear that depth alone is not sufficient to deal with class imbalances and other domain difficulties. This is clear from the number of low bars or blank regions in the graphs of Figs. 9 in the main body of the paper and 18 in the appendix.

6.3 Answering RQ3: does regularization modify the equation?

The current guidance on DL is to also use regularization techniques. This is what we did in this section to see whether such techniques offer any novel insights. Figure 10 shows the optimal settings reported in this section for MLP on the Backbone domain.

Backbone domain Comparing the results obtained with and without regularization (See Figs. 24 and 25 in the appendix for sizes 5 and 1 with regularization, respectively), we see that the improvement is often negligible. For $s = 5$, on the large version of the backbone domain, we found that the only non negligible improvement brought upon by regularization (Early Stopping alone in this case) was an improvement in G-Mean of about .05 for concept complexity $c = 5$, balanced level $b = 5$ and depth $d = 5$. For $s = 1$, the small version of the backbone domain, there were 6 cases where a non-negligible improvement of .1 or .2 in G-mean was observed. However, in all these cases, regularization brought the performance from 0 to .1 or .2, meaning that the results remained non-competitive despite the improvement. In one case, ($b = 4, c = 4, d = 2$), the .1 improvement was a bit more meaningful, however, as the performance was brought to around .7. In two other cases where the results were already close to 1 prior to regularization, the improvement was negligible. In the small size domain, the type of regularization used were diverse combinations of the three types considered.

Image domain The regularization experiments were not conducted in the text domains given the little advantage they displayed in the Backbone domain.

□ *RQ3 for MLP answered* Altogether, while regularization seems able to help improve the performance on some domains, these improvements remain rather limited and do not have much practical significance in the context of concept complexity, class imbalance and data scarcity.

¹⁰ For confirmation, we ran a separate experiment with 10 layers on the 20 Newsgroup domain and saw even greater deterioration of the results.

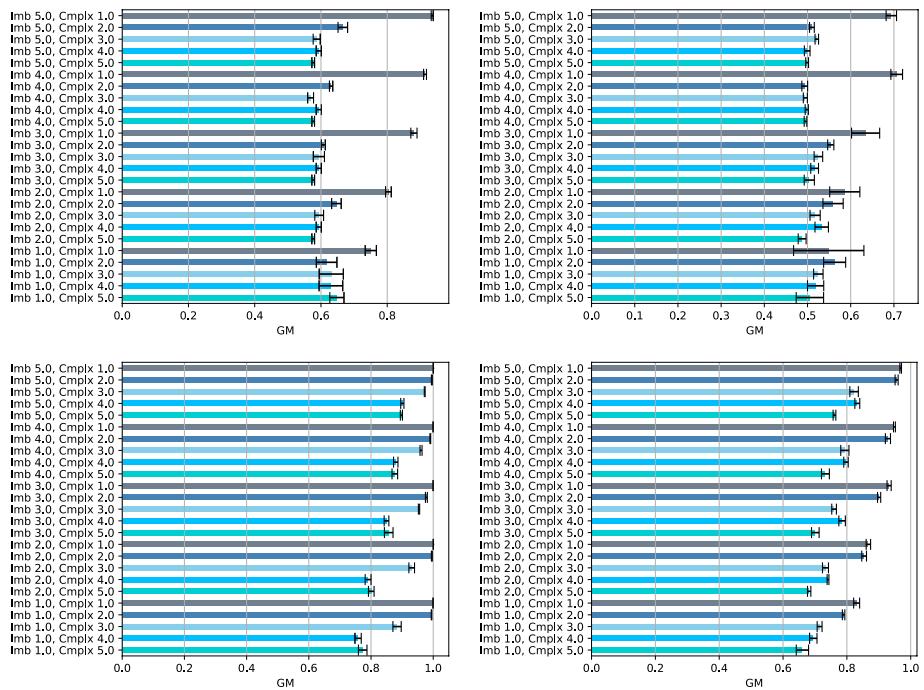


Fig. 11 G-Mean results of applying deep CNN Models to the Artificial Shape domains. (Top Left: Large Data Sets; Top Right: Small Data Sets; Bottom Left: MNIST-Fashion; Bottom Right: Cifar-10; Low complexity: level 1; High complexity: level 5; High imbalance: level 1; Low imbalance: level 5.)

7 Results in the convolutional neural networks case

Overview We now discuss the results obtained with CNN in detail. We first discuss the dependency between class imbalances, concept complexity and sample size; we then test whether the depth of the networks used makes a difference; and we conclude with an analysis of the role of regularization on the results.

Summary of main observations Our observations for CNNs differ from ones for deep MLPs. While class imbalance still negatively affects CNNs, the impact of this learning difficulty is smaller than for deep MLPs. However, data size (or data sparsity) is always a big challenge for CNNs. Increasing CNNs' depth does not lead to gains in robustness or predictive power, an observation dramatically different from previously reported study (Ding et al., 2017). Finally, regularization techniques once again did not have a significant impact on the robustness to class imbalance.

7.1 Answering RQ1: What is the dependency between class imbalance, concept complexity, sample size and deep learning?

The results obtained by the CNNs are shown in Fig. 11 for the shape and image domains. Here again, we discuss to what extent each of the four observations made in Japkowicz and Stephen (2002) translate to the deep CNN settings.

⇒ *Observation 1 In traditional learning approaches, linearly separable domains are not sensitive to class imbalance*

Shape domains While, in the shape data set, we cannot truly distinguish between linearly and nonlinearly separable domains given the nature of the complexity introduced in the shape data set, we can distinguish between easier and more complex problems with regard to the size of the separation margin. We observe that in the simplest kind of shape problem, that of complexity level $c = 1$ (star versus square), the only one that can be truly be handled by the CNN, the class imbalance problem is harmful. Indeed, we observe a decrease in performance, as the balance level decreases, from about .95 to below .8 for the large data set size ($s = 5$) and from about .7 to .55 for the small data set size ($s = 1$). Therefore, the least complex domain is not spared from the class imbalance problem.

Image domains In the MNIST Fashion data set, the imbalance level does not affect the performance of the CNN on the two simplest domains ($c = 1$ and 2). In the CIFAR-10 domains, on the other hand, even on the simplest domains, a small decrease in performance from about .95 to below .8 is observed as the balance level drops from $b = 5$ to $b = 1$.

⇒ *Observation 2 In traditional learning approaches, as the class imbalance increases, so does the amount of misclassification*

Shape domains In deep CNN, the same trend is observed for concept complexity $c = 1$ and to a very small extent, concept complexity $c = 2$. It is not observed in any of the other concept levels because the problems are just too difficult to be solved by our CNN and the low performance hovering around .6 is relatively constant for all class imbalance levels.

Image domains In MNIST-Fashion, the same trend is encountered, but only on the more complex domains, starting at concept complexity $c = 3$. The decrease in performance, however, is very slight. For example, for concept complexity $c = 5$, the performance drops from about .9 to slightly below .8. In CIFAR-10, the trend is encountered for all concept complexity levels. The drop is actually more pronounced in simple domains than in more complex ones. Indeed, for concept complexity $c = 1$, performance drops from about .95 at balance level $b = 5$ down to about .82 at balance level $b = 1$. Yet it drops from about .75 at balance level $b = 5$ to about .65 at balance level $b = 1$ for concept complexity $c = 5$.

⇒ *Observation 3 In traditional learning approaches, the class imbalance problem is exacerbated by the problem's complexity*

Shape domains In deep CNN, it is not clear that this trend is observed. In the shape domains, the class imbalance problem is visible in concept complexity $c = 1$, for both data set sizes, as previously observed. It is very slightly present for concept complexity $c = 2$ in the large data set setting, and is not present at all for any other concept complexity level and data set size. So, it cannot be said, based on this data set that the class imbalance problem is exacerbated by the problem's complexity. On the contrary, it becomes less and less pronounced, but we suspect that that is due to the low performance attained by CNN on these more complex domains.

Image domains On the MNIST Fashion data, concept complexity has an impact on the effect of the class imbalance problem. Indeed, at concept complexity $c = 1$ and $c = 2$, the networks are not affected by class imbalance. They are slightly affected at concept complexity $c = 3$, and more so at concept complexity $c = 4$ and 5. Overall, however, the negative effect is relatively slight since the largest effect recorded is at concept complexities $c = 4$ and 5, where the performance goes from about .9 at balance level $b = 5$ to a bit below .8 at balance level $b = 1$. In the CIFAR-10 domain, the degree of concept complexity does not aggravate the loss of performance caused by class imbalance much since the rate of decrease in performance appears to be more or less the same for all concept complexity level as the class imbalance increases, and as mentioned earlier, the decrease may be larger

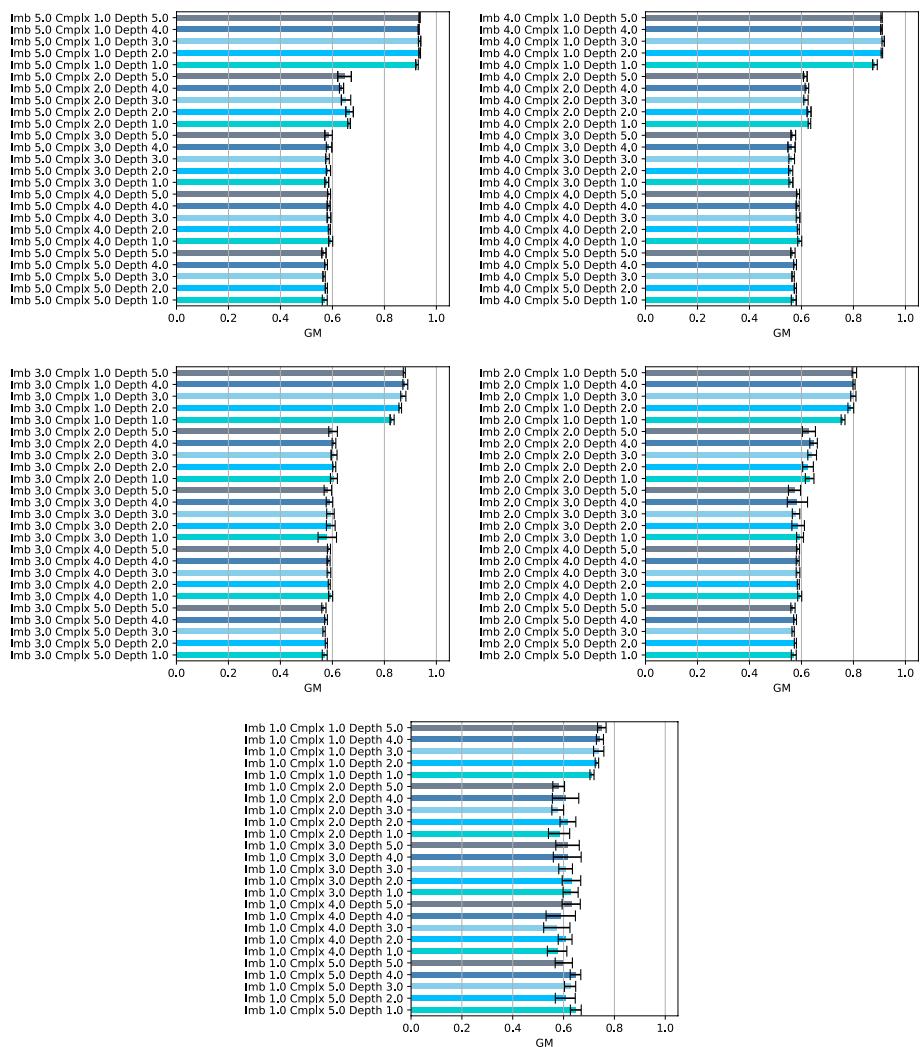


Fig. 12 The effect of depth on class imbalance levels, complexity in large datasets in the Shapes domain. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

for the case of concept complexity $c = 1$ than it is for concept complexity $c = 5$, so in fact class imbalance may affect the least complex domains more than the more complex ones on this domain.

⇒ *Observation 4 In traditional learning approaches, the class imbalance problem is exacerbated by data scarcity*

Shape domains This trend is also observed on the shape data set. To begin with, the performance of the CNN is lower by about .2 in the smaller data set size ($s = 1$) than in the larger one ($s = 5$). The decrease in performance also seems steeper with respect to a drop in class balance in the case of the smaller data set than in the case of the larger one. This is

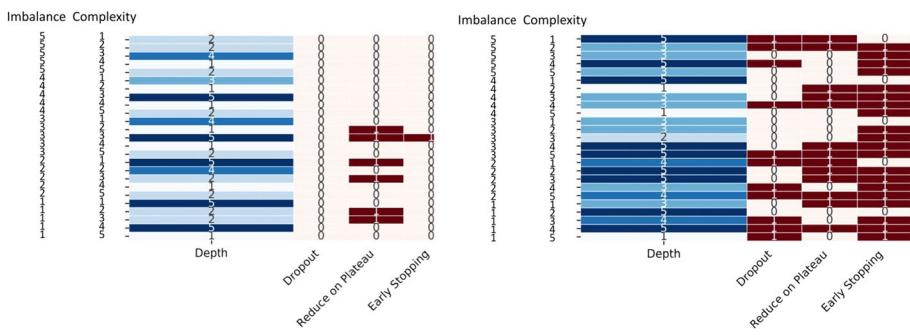


Fig. 13 Optimal Depth and Regularization settings for CNN experiments on Shapes Data Sets for large data set size ($s = 5$, left) and small dataset size ($s = 1$, right)

shown both by the height of the bars and the degree of uncertainty displayed by the standard deviation line which is quite wide at balance levels $b = 1$ and 2 in the case of the small data set size.

Image domains This trend was not tested in the image domain.

□ *RQ1 for CNN answered* All in all, this analysis concludes that CNN do not behave in exactly the same way as their MLP counterparts. While they are affected by the class imbalance problem, they are less so than MLP, and concept complexity, at least concept complexity related to the size of the separating margin, does not appear to be as significant a factor as it is in the MLP case. That being said, the type of complexity considered in the MLP case is closely related to the question of data set size and small disjuncts and since CNN are very sensitive to data set size as well, there may be a closer connection between the two types of networks than our analysis may suggest.

7.2 Answering RQ2: how does the depth of the networks affect the imbalance/complexity/performance triangle?

Shape domains Figure 12 suggests that the depth of CNN has mostly no effect on their performance no matter what complexity level or degree of balance in the large data set of size $s = 5$, except for the fact that at balance level $b = 1$, the performance as a relation to depth becomes unstable. In other words, the deeper the network, the less stable, in such cases. The same is true for the data set of size $s = 1$, as shown in Fig. 21 in the appendix, except for the fact that the instability starts as early as at balance level $b = 3$.

Image domains Figures 22 and 23 in the appendix show a significant decrease in performance in the MNIST-Fashion and CIFAR-10 domains when the CNN are very deep such as depth $d = 5$, and sometimes $d = 4$ or 3 , in certain cases. This suggests that the networks may be overfitting the data, or that, perhaps, their capacity is too large and they have been undertrained. The reason for this observation needs to be investigated further.

□ *RQ2 for CNN answered* Depth does not seem to play as important a role in the case of CNN than in MLP networks, where depth was not sufficient to handle class imbalances, but was shown to help in some cases. In the case of CNNs, depth does not help with either class imbalances, concept complexity or data scarcity. In fact, our experiments show that too deep a network may, in fact, be harmful.

7.3 Answering RQ3: Does regularization modify the equation?

Shape domain Comparing the results obtained with and without regularization (see Figs. 26 and 27 in the appendix for sizes 5 and 1 with regularization, respectively), we see that the improvement is often negligible. For $s = 5$, on the shape domain, we found that in all six cases where regulation helped, the improvement in G-Mean was negligible (around .01 or .02). For $s = 1$, regularization methods were invoked in almost every case. Figure 13 displays the optimal depth and regularization on the Shapes datasets for small and large dataset size. However, as in previous instances, the improvement they brought upon was often negligible or very small and not terribly significant. For example, in the cases of $b = 1$ or 3 and $c = 4$, the G-Mean went from around .5 to close to .6.

Image domain The regularization experiments were not conducted in the image domains given the little advantage they displayed in the Shape domain.

□ *RQ3 for CNN answered* Similarly as in the case of MLP, applying regularization methods for CNN offer highly limited improvements remain and do not have much practical significance in the context of concept complexity, class imbalance and data scarcity.

8 Lessons learned from experimental investigations

The purpose of our experiments was to observe the combined effect of class imbalances and other domain characteristics on deep learning domains, in order to better understand the roots of the problem and subsequently find robust solutions that address the problem in a deep rather than superficial manner. The results shown and analyzed in detail in the previous two sections are summarized here in terms of their main takeaways.

Lessons learned for MLPs Firstly we present a list of our main observations pertaining behavior of MLP models:

- Deep MLP models, like their shallow counterparts and other traditional classifiers, are deeply affected by concept complexity, class imbalances and data size (sparsity). That is shown in both artificial and practical data sets.
- There are some conditions of imbalances and other complexities where a depth of 1 and sometimes even 2 is not sufficient for MLP to handle the problem. In these cases, increasing their depth is useful. The effect of such increases when called for, however, tends to plateau after a sufficient depth is found, and in certain cases, the performance, actually decreases, with too large a depth. This last observation is on par with the results obtained by Johnson and Khoshgoftaar (2020) on specific data sets and may help explain the effect they noticed. Altogether, our results lead us to conclude that depth should not be discarded: it can have a beneficial (as well as a detrimental) effect on the class imbalance problem for MLPs, but this effect is limited to certain situations and the increase in depth may, in fact, become a hindrance.
- Regularization techniques such as Dropout, Reduce on Plateau and Early stopping typically do not have a significant effect on class imbalances, concept complexity and data size (sparsity). They sometimes help raise performance slightly above zero, but that improvement is of little practical value. In rare cases, they help slightly with more practically relevant results.

Lessons learned for CNNs Interestingly, our observations for CNN models differ from the MLP ones:

- We begin with a general observation independent of the class imbalance problem: CNN are not that appropriate for the raw shape domains. While they can adequately distinguish between a star and a square, they are unable to distinguish between different types of polygons. This suggests that CNN may not work well on domains where the separating margin is very small. On the other hand, CNN are much better suited to the MNIST-Fashion and CIFAR-10 tasks.
- Class imbalances have a negative effect on the performance of CNN, though that effect seems smaller than it is in the MLP.
- It is not always clear that the negative effect caused by class imbalances is more pronounced in complex domains than in less complex ones.
- Data size affects CNN's performance quite significantly and independently of class imbalances.
- Increasing the Depth of CNN does not have any effect on their performance, except for a sharp decrease in performance, in certain instances of very large depth. This is a bit different from what was reported in Ding et al. (2017) who found that increasing the depth from 6 to 10 helped improve performance but didn't help, with performance, when increased beyond 10. This may be because the domains in Ding et al. (2017) were more complex than the problems considered here. That may also explain why an increase in depth caused a decrease in performance in our experiments while it didn't in Ding et al. (2017). This decrease in performance with depth, by the way, is consistent with the results obtained with MLP.
- Regularization techniques such as Dropout, Reduce on Plateau and Early stopping typically do not have any kind of significant effect on class imbalances, concept complexity and data size.

9 Literature review II: addressing the class imbalance problem in deep learning systems

The results of our experiments generally confirm the findings of the literature discussed in Sect. 2 and provide a discussion of the detailed situations where class imbalances are more or less harmful and how depth and regularization affect these findings. All in all, it is clear that the class imbalance problem has not been erased by deep Learning and that solutions need to be found to counteract them. The purpose of this section is to present some of the solutions that have been proposed to date to address the problem. The review of these solutions will provide a better understanding of the state-of-the-art solutions for tackling the imbalance problem in the context of deep learning. Moreover, this section will help us to better understand the open issues and challenges in this domain which will be presented and discussed in Sect. 10.

Overview A large volume of solutions has been proposed for tackling the class imbalance problem. These solutions can be categorized into pre-processing approaches, special purpose algorithms or post-processing approaches (Branco et al., 2016; Krawczyk, 2016). In the particular context of deep learning, the class imbalance problem only recently started to be addressed and several approaches on the three possible categories have been proposed and tested. However, important requirements must be considered when applying

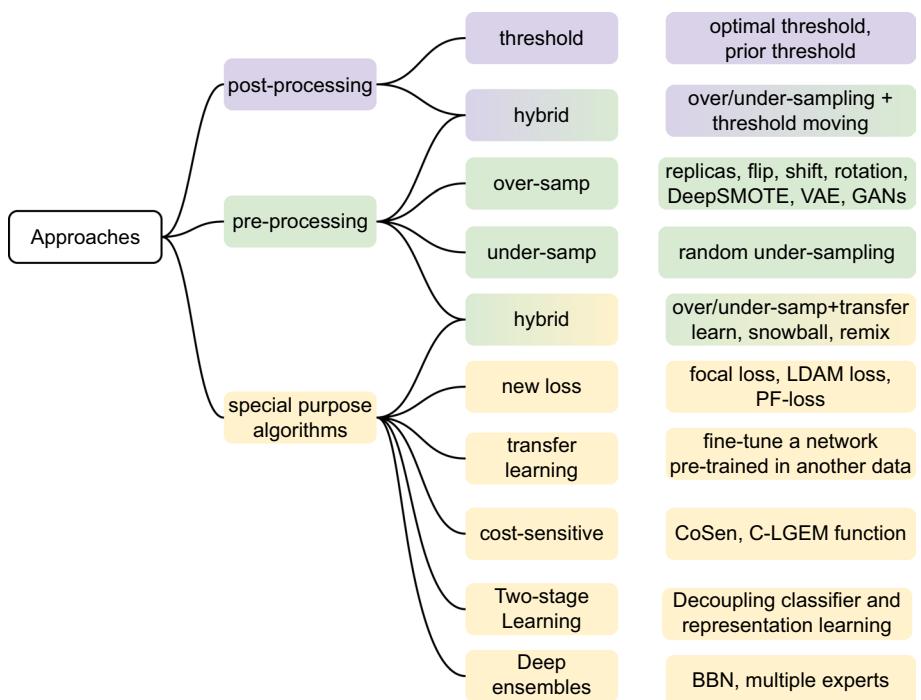


Fig. 14 Three main categories of approaches proposed and tested for tackling the class imbalance problem when using deep learning methods. Main categories of approaches on the left, followed by subcategories and some examples on the right

previously developed approaches due to the fact that many of deep learning methods investigated are developed specifically for dealing with images instead of tabular data. This is the case when applying more sophisticated pre-processing (resampling) approaches that generate new synthetic cases through the interpolation of two cases. These solutions need to be applicable to raw images and ideally should preserve the images characteristics. This means that the methods used for generating new synthetic examples should be able obtain meaningful, high-quality images that can be visually inspected and that can improve the discriminative power of deep learning (Dablain et al., 2022).

Taxonomy of deep learning solutions to class imbalance The main approaches for tackling class imbalance that have been proposed and tested with deep learning methods are summarized in Fig. 14. The post-processing approaches act only on the predictions, while the pre-processing approaches are applied on the training stage and the special purpose algorithms are connected to the learning stage.

Review of existing approaches Changing the decision threshold using prior class probabilities is a pure post-processing approach that was used in deep learning with images. Buda et al. (2018) explored this approach which showed performance advantages when compared against the baseline models. This approach can be easily implemented and allows to obtain adjusted predictions without retraining the model. In effect, thresholding helps to determine a threshold on the network outputs that reduces the probability of misclassifying minority class cases. Buda et al. (2018) also tested

a hybrid variant integrating both pre-processing and post-processing approaches that combines random under-sampling or random over-sampling with a threshold adjustment of the predictions. Overall, thresholding applied jointly with over-sampling exhibited good results in the experiments carried out. Johnson and Khoshgoftaar (2021) explored two thresholding approaches with MLPs: the optimal threshold, which uses validation/training data to determine the threshold that maximizes the geometric mean; and the prior threshold, which sets the threshold to the prior probability of the minority class. The statistical test results showed that there is no difference between both approaches. These approaches are better than the default threshold of 0.5 that only displays good performance in fairly balanced scenarios.

As previously mentioned, pre-processing approaches have some restrictions when the predictive task involves images. The majority of the works that have explored over-/under-sampling approaches used CNNs as the base architecture and restricted their evaluation to the following: random under-sampling the majority class, random over-sampling of the minority class by adding replicas, flips, shifts, rotations, shear or changing the brightness (e.g. Hensman and Masko (2015); Buda et al. (2018); Pouyanfar et al. (2018); Johnson and Khoshgoftaar (2020)). Overall, the experiments indicate that over-sampling approaches have a better performance when compared to under-sampling (Buda et al., 2018; Johnson & Khoshgoftaar, 2020). Recently, DeepSMOTE was proposed as a strategy that integrates the SMOTE algorithm with deep learning methods (Dablain et al., 2022). This method overcomes the difficulties of data generation enabling its application in tabular and image domains. In order to generate new synthetic examples GANs have been a popular over-sampling alternative for image datasets (Sampath et al., 2021). Multiple GAN architectures such as, WGAN-GP (Bhatia & Dahyot, 2019), CGAN (Nazari & Branco, 2021) or BAGAN (Mariani et al., 2018), have been explored to achieve the generation of synthetic data. The majority of the works using GANs for synthetic data generation in class imbalance scenarios are focused on image generation and use medical image datasets. Variational autoencoders (VAEs) (Kingma & Welling, 2013) and Wasserstein autoencoders (WAEs) (Tolstikhin et al., 2017) have also been used as methods for synthetic cases generation.

Hybrid approaches that combine pre-processing and special purpose algorithms have also been tested. For instance, a dynamic sampling technique that combines under and over-sampling methods with transfer learning was proposed by (Pouyanfar et al., 2018). Lee et al. (2016) pre-trains a CNN with class normalized data obtained by applying random under-sampling. Then, the CNN is fine-tuned (retrained) using the original training data. Murphrey et al. (2004) carries out experiments with snowball training (Wang & Jean, 1993). This technique starts by training a neural network exclusively with the minority class examples. Then, the network capability to recognize the majority class examples is increased by using a dynamic training set that contains all minority class examples and an increasing number of majority class examples. Buda et al. (2018) uses a two-phase learning method that starts by training the network on a balanced data and then the output layers are fine-tuned.

A promising hybrid approach concerns the use of batch adjustments. These include applying a balancing strategy in the batches which can be achieved, for instance, through oversampling or prioritized sampling. Mullick et al. (2019) discusses the fact that applying SMOTE in the context of deep learning may not be appropriate due to the fact that deep learning carries out an end-to-end process that integrates both deep feature extraction and classification. This makes it difficult to incorporate oversampling techniques which are usually applied after the feature extraction process. Moreover, frequently the datasets

are very large to load and pre-process. This motivates the application of resampling at the batch level in deep learning to balance the training distribution. Dong et al. (2017) propose a batch-wise incremental hard mining of hard-positives and hard-negatives that considers exclusively the minority attribute classes. This approach is supported by a Class Rectification Loss (CRL) regularising algorithm. CRL is designed to improve the minority class learning in each batch with batch-balance updated deep features. Bellinger et al. (2021) propose ReMix, a training technique that combines batch resampling, instance mixing and soft-labels to improve the performance on imbalanced datasets. ReMix uses a prioritized mixing strategy for efficiently balancing the classes in each training batch while increasing the minority class space. Several regularization approaches specifically designed for imbalanced deep learning have been proposed. Examples of such solutions include the Major Feature Weakening (Ye et al., 2021) and Remix (Chou et al., 2020). The Major Feature Weakening method is a new learning strategy that seeks to equalize the training progress across the different classes of the problem. The key idea of this method is to mix features of the majority class with those from other data in a mini-batch which weakens these features, and thus prevents the neural network from fitting them first.

Ren et al. (2018) propose a method that also works on the batch level, where the gradient direction is used to learn and assign weights to the training cases. The work of Sellami and Hwang (2019) presents a dynamic batch-weighted loss function for heartbeat classification. In this solution the loss weights change in a dynamic way as the distribution of the classes in each batch changes. Other solutions involving the notion of batch adjustments have been proposed more recently [e.g. Hu et al. (2021), Peng et al. (2021)].

Cross Entropy (CE) loss is the loss function most frequently used loss function for deep learning. However, this loss assumes that the training dataset is balanced and thus is not the ideal setting when tackling the class imbalance problem through deep learning. Several alternative loss functions have been proposed to deal with this issue. For instance, the WCE is a variant of the CE loss that assigns a higher weight to samples from the minority class and a lower weight to majority class samples. Wang et al. (2016) proposed two new loss functions, the Mean False Error (MFE) and the Mean Squared False Error (MSFE), that have a higher sensitivity to minority class errors. The focal loss proposed by Ross and Dollár (2017) is another loss function that aims at changes the CE loss for reducing the impact on the loss of easy to learn examples. The CE loss is thus multiplied by a new factor while the class weight used in WCE is also kept to allow the loss to not focus on hard to learn majority class samples. Other loss functions have been presented to address other aspects. For instance, the Label-Distribution-Aware Margin loss (LDAM) (Cao et al., 2019) was proposed for encouraging larger margins on minority classes and the Parameter Free-loss (PF-loss) proposed by Du et al. (2021) tackles the issue of parameter tuning by removing them from the equation. In the PF-loss, no hyperparameters exist and the loss dynamically adapts to give more attention to the minority class. The PF-loss presents multiple advantages including a reduced training time, a focus on the minority classes instead of the outliers, and an increased accuracy due to its adaptation to different classes distributions. Transfer learning is an alternative special purpose method that begins with pre-training the network on a dataset different from the one targeted in the predictive task, and then fine-tunes the network on the task dataset (Pouyanfar et al., 2018).

Cost-sensitive is a know method for dealing with the class imbalance problem. In the context of deep learning, several approaches have been put forward for embedding cost-sensitive learning. For instance, Wang et al. (2018) modified the CE loss function used by an MLP by integrating a pre-defined cost matrix. This will force the to focus on the minority class in order to minimize the misclassification cost. The minimization of a new

cost-sensitive localized generalization error-based objective function (C-LGEM) is proposed by Ng et al. (2021). The C-LGEM function is used in the neural network training to achieve a better balance between the errors of the majority and minority classes. Specific cost-sensitive solutions were proposed for deep learning in certain imbalanced tasks. Fuqua and Razzaghi (2020) and Geng and Luo (2018) present examples of such methods developed for convolutional neural networks targeting the chart pattern recognition problem and time series classification, respectively. The CoSen CNN proposed by Khan et al. (2017) follows a different approach that allows learning the cost matrix and the network weight parameters simultaneously. This solution has the advantage of not requiring the end-user to define the cost matrix, a time-consuming and costly task.

A two-stage learning approach was proposed for deep imbalanced learning by Kang et al. (2019). The authors found that pre-processing strategies are relevant when jointly learning representation and classifiers in imbalanced domains. However, decoupling the representation learning and classifier learning has advantages. The authors found that class imbalance might not be a problem for representation learning with high-quality representations being obtained through the simplest instance-balanced sampling. Given the strong representation, state-of-the-art results are obtained through re-balancing the classifier. This research gave rise to other works that developed deep ensembles which focus on the decoupling of the representation and classifier (Zhou et al., 2020; Wang et al., 2020a). An example of such solutions is the Bilateral-Branch Network (BBN) (Zhou et al., 2020). The BBN model uses two branches that tackle the representation learning and classifier learning simultaneously and separately. A cumulative learning strategy is also embedded to allow to first learn the universal patterns giving attention to the tail data later on and in a gradual fashion.

10 Open challenges for class imbalance in deep learning

Research in this field is still in its initial steps, and several challenges remain open when handling class imbalance through deep learning methods. Below we list and discuss most vital challenges and future directions for this domain:

1. *More efficient instance generation for oversampling* There is only a limited number of approaches for generating synthetic instances and standard interpolation methods for synthetic data generation are not generally applicable to image data. Moreover, in the deep learning context it is typically not possible to load the entire dataset which represents an added challenge; thus, resampling strategies may be applied in mini-batches, which brings limitations in terms of the quality of the cases generated and the complexity of applying the strategies. In effect, auxiliary computations necessary for synthetic data generation, such as calculating nearest neighbors in each mini-batch, may result in a slower training which may have severe consequences for large problems.
2. *Oversampling beyond images* There is a need for a holistic oversampling approaches, capable of working with various modalities of data. While there exist highly effective oversampling solutions, like DeepSMOTE (Dablain et al., 2022) or GAMO (Mullick et al., 2019), they are mainly dedicated to image representations. Solutions for other data modalities exist. However, a general solution suitable for any type of data is still missing. One can envision a more general approach capable of generating of artificial instances of any type of complex representations, such as sounds, text, videos, etc.

3. *Rigorous and standardized test bed evaluations* Many resampling approaches are based on GANs (or other generative models), which still lack standardized approaches for evaluation of their quality. The imbalance problem is thus an added challenge. Importantly, we need to understand what is the relationship between the visual quality of generated images and their actual impact on the class imbalance problem. Furthermore, we need rigorous and reproducible evaluation measures for their capability of generating information-rich and out-of-training-distribution instances that would truly enrich minority classes.
4. *Analysis of the imbalanced deep learning toolbox* Theoretical and empirical comparisons of cost-sensitive learning and resampling in the traditional class imbalance literature, such as Drummond and Holte (2003); Weiss et al. (2007); Seiffert et al. (2008a, 2008b); Wallace et al. (2011), have served to advance our understanding of when, where, and why these methods might work. Deep learning, however, presents challenges and opportunities that did not exist at the time of these studies. As a result, it still remains unclear which imbalanced learning tools should be used in deep learning. A thorough understanding of the relationships between the solutions for the imbalance problem and their impact in a deep learning context is still lacking. Deep generative methods, for example, have the potential to generate more realistic synthetic samples to represent the minority class Mullick et al. (2019); Wang et al. (2020b). On the other hand, it has been argued that in large data regimes, the benefit of adding new samples to imbalanced classes is diminished Cui et al. (2019). Re-weighting has demonstrated some promise as it can easily be applied on a per-class or per-sample basis. However, doing so may negatively impact optimization Huang et al. (2016, 2019a). In these cases, a class imbalanced loss function maybe helpful Cao et al. (2019); Cui et al. (2019); Li et al. (2019); Lin et al. (2017). A comprehensive analyses of the tools for imbalanced deep learning, such as those undertaken in the traditional setting, will be of great benefit to guide researchers and practitioners as deep learning continues to expand into real-world applications.
5. *Mapping between domain complexity characteristics and deep learners' effectiveness* As observed in the literature review presented in this paper, the relationship between the domain complexity and its impact in deep learning systems is still not well understood. An initial step to address this issue would be to provide a more formal and systematic definition of different complexity characteristics which could then be analysed and connected to the effectiveness of different deep learning systems. Some studies already address this problem when observing the relation between class overlap its impact on the performance of deep learning methods (Ghosh et al., 2021). However, a more thorough and complete understanding of the relationship between multiple complexity characteristics of the tasks and their impact in deep learning models is still needed. This challenge is also presented by Santos et al. (2022), for the particular case of class overlap, where the need to develop a well-established formulation and measurement of class overlap in the context of real world problems is put forward.
6. *XAI for class imbalance* So far there is almost no research on Explainable Artificial Intelligence (XAI) approaches tailored to class imbalance. Such models would allow us to gain a deeper insight into the nature of class imbalance in deep learning, allowing for understanding what type of information impacts in what way both the feature extraction and classification. This would be especially useful for building trust with the medical and health science communities that deal with a plethora of imbalanced problems, but display scepticism towards many skew-insensitive techniques, especially based on artificially generated instances.

7. *Class imbalance in continual and lifelong learning* Continual and lifelong learning are rapidly developing domains of deep learning, focusing on creating adaptive models. They hold a lot of similarities to data stream mining domain, as they all deal with continually arriving new information and could be seen as two sides of the same coin (Korycki & Krawczyk, 2021). Class imbalance poses unique challenges in such a setup, as deep learning models must be capable of overcoming catastrophic forgetting and emerging imbalance ratio that may come with incremental tasks or classes. Additionally, in such setting there will be another bias towards the most recent task or class, leading to models being skewed on multiple fronts. The most challenging scenario would arise with addition of concept drift (Guzy et al., 2021), where previously seen classes may evolve and the imbalance ratio among classes will change over time.
8. *Deep learning from imbalanced time series* Learning from imbalanced time series is a very challenging task, as the imbalance may appear either as the number of training instances, or as rare occurrence of specific events or characteristic in the series (Krawczyk, 2016). Despite massive success of deep learning methods for classifying and forecasting time series, there are but few methods dedicated to handling skewed problems (Huang et al., 2019b). There is a need for creating deep oversampling approaches capable of generating not only life-like time series, but also their specific segments. Furthermore, forecasting methods must be enhanced with skew-insensitive mechanisms in order to allow for unbiased predictions, as well as understanding what specific characteristics of time series pose a challenge to deep models.

11 Conclusion

The purpose of this paper was to seek a better understanding of the state of the art in deep learning in relation to the Class Imbalance problem. It considered the matter from two standpoints.

Summary of the literature review First, it proposed a detailed literature review of the work that has been done on the topic. It particularly looked at two sub-areas of research: the effect of class imbalances on deep learning; and the way in which the class imbalance problem has been addressed to date in deep learning systems. From this review we confirm that deep learning methods are affected by the class imbalance, and we provide evidence of the difficulties these methods face. We verified that assessing the predictive task complexity is difficult and has not been sufficiently explored. Regarding the strategies for handling class imbalance in deep learning we observe that several attempts have been made in the three categories of approaches. However, these methods still present important limitations including: limited number of solutions; difficulty in applying standard interpolation methods to image data; restriction of the application of resampling methods to mini-batches; and other computational efficiency issues associated to standard resampling approaches.

Summary of the controlled experiments Second, the article ran a number of controlled experiments on both artificial and real domains, using MLP and CNN of various depths in order to establish some basic and systematic understanding of how deep learning responds to Class Imbalances. The idea was to create the same kind of basic understanding as that available for traditional learning systems. It was found that the deep MLP were affected by the class imbalance problem and the associated issues of concept complexity and data scarcity, in ways very similar to the way in which traditional learning systems are. While depth helped lessen the problem in certain cases, regularization did not help much. This suggests

Fig. 15 Variability of the square class

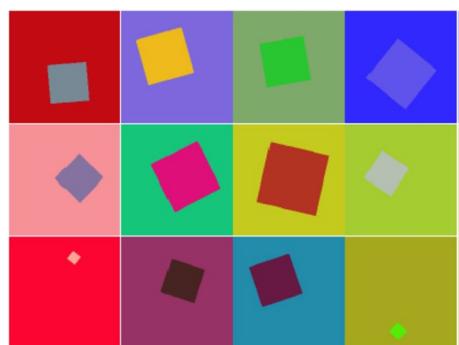


Fig. 16 Illustration of the modified Shape images

that, like in the traditional learning setting, novel solutions need to be proposed. The deep CNN were affected by the class imbalance problem, but somewhat less so. On the other hand, data scarcity seemed to have a much greater negative impact on them. Once again, neither depth nor regularization were sufficient to combat the problems and novel solutions are needed.

Proposed future work This study suggests a number of future avenues to explore. First, it would be interesting to generate new domains presenting different kinds of concept complexities and to run experiments on a greater variety of real domains. This can be done using the measure of complexity previously proposed for class imbalances and complexity such as Barella et al. (2021). Although this study has taken steps to reduce the risk of vanishing gradients, the problem cannot be fully eliminated in deep learning. An important next step is to undertake a detailed study of the relationship between class imbalance and the vanishing gradient problem. It would also be interesting to expand our work to multiclass classification problems to confirm and explain in more detail the results of the different studies that have been conducted on this category of problems. Second, an important family of deep learner we omitted from this study is the Generative Adversarial Network. It would be useful to extend the study to include such networks. Of course, another avenue is to use the insights gained by this study to create new approaches to deal with the class imbalance problem.

complexity	imbalance	Majority Class Label	Minority Class Label	Majority Class Label Name	Minority Class Label Name	Class Distribution
1	1	4	1	Marketing Pharmaceutical	Data Management and Statistics	829 : 21
1	2	4	1	Marketing Pharmaceutical	Data Management and Statistics	829 : 44
1	3	4	1	Marketing Pharmaceutical	Data Management and Statistics	829 : 146
1	4	4	1	Marketing Pharmaceutical	Data Management and Statistics	829 : 355
1	5	4	1	Marketing Pharmaceutical, Healthcare and	Data Management and Statistics	829 : 829
2	1	5	7	Medical Sales Pharmaceutical, Healthcare and	Regulatory Affairs	1163 : 30
2	2	5	7	Medical Sales Pharmaceutical, Healthcare and	Regulatory Affairs	1163 : 61
2	3	5	7	Medical Sales Pharmaceutical, Healthcare and	Regulatory Affairs	1163 : 205
2	4	5	7	Medical Sales	Regulatory Affairs Pharmaceutical, Healthcare and	1163 : 498
2	5	7	5	Regulatory Affairs	Medical Sales Pharmaceutical	1163 : 1163
3	1	0	4	Clinical Research	Marketing Pharmaceutical	1071 : 27
3	2	0	4	Clinical Research	Marketing Pharmaceutical	1071 : 56
3	3	0	4	Clinical Research	Marketing Pharmaceutical	1071 : 189
3	4	0	4	Clinical Research Pharmaceutical	Marketing	1071 : 459
3	5	4	0	Marketing	Clinical Research	1071 : 1071
4	1	0	7	Clinical Research	Regulatory Affairs	1163 : 30
4	2	0	7	Clinical Research	Regulatory Affairs	1163 : 61
4	3	0	7	Clinical Research	Regulatory Affairs	1163 : 205
4	4	0	7	Clinical Research	Regulatory Affairs	1163 : 498
4	5	0	7	Clinical Research	Regulatory Affairs Manufacturing & Operations	1163 : 1163
5	1	8	2	Science		2235 : 57

Fig. 17 Number of training instances in the Job Classification domain

Appendix

We now present additional figures and plots that provide more details on the results discussed in this paper.

A Detailed illustration of the shapes domain

Figure 15 shows the variability found in the squares class. As shown in the figure, the images may differ in terms of background color, square size, position and orientation. While this is illustrated in the case of squares, the same is true for all the other shapes considered in this domain. Figure 16 shows the results of the shape images after they have been modified as discussed in the paper (Fig. 17).

B Instance numbers used in the job classification domain

We now present the number of instances used in all settings of the job classification data set. These numbers vary from concept complexity to concept complexity and could not be summarized in a table like the other domains.

At each complexity level the testing set is made of the majority number of instances for each class. For example, the testing set for complexity level 1 is 829 instances of class 1 and 829 instances of class 2; for complexity level 2, it is 1,163 and 1,163; and so on.

C Experimental setup

For each of the MLP networks, we report the optimal results recorded after running them with 2, 4, 8, and 16 hidden units in each layer. The number of hidden layers in the network is increased from 1 to 5. We trained each of the MLP networks for 300 iterations, with a learning rate of 0.001, and using the Adam optimizer. Relu activation and uniform weight initialization were utilized as recommended in Glorot and Bengio (2010) to reduced the risk of vanishing gradients. We also experimented with three different types of regularization approaches: Dropout, Reduce on Plateau and Early stopping.

We evaluated the CNN networks, with an increasing number of convolutional layers (filters): 1 (8), 2 (8-16), 3 (8-16-32), 4 (8-16-32-64), 5 (8-16-32-64-64). Two dense layers are featured at the end of each model architecture. Relu activation and uniform weight initialization were used in these models (Glorot & Bengio, 2010). We also experimented with three different types of regularization approaches: Dropout, Reduce on Plateau and Early stopping.

D Additional results on depth for multi layer perceptrons

This part of the appendix provides additional results pertaining to the effect of Depth on Multi-Layer Perceptrons in the cases of class imbalances, concept complexity and data scarcity. These results were not included in the main body of the paper to improve the presentation of the results. However, we felt that it was important to include them for the interested reader. Figure 18 shows the results obtained on all depths for the Backbone Domain of small size ($s = 1$). Figures 19 and 20 show the results obtained on all depths on the text domains, namely, 20 Newsgroups and Job Classification. These results are discussed in the body of the paper.

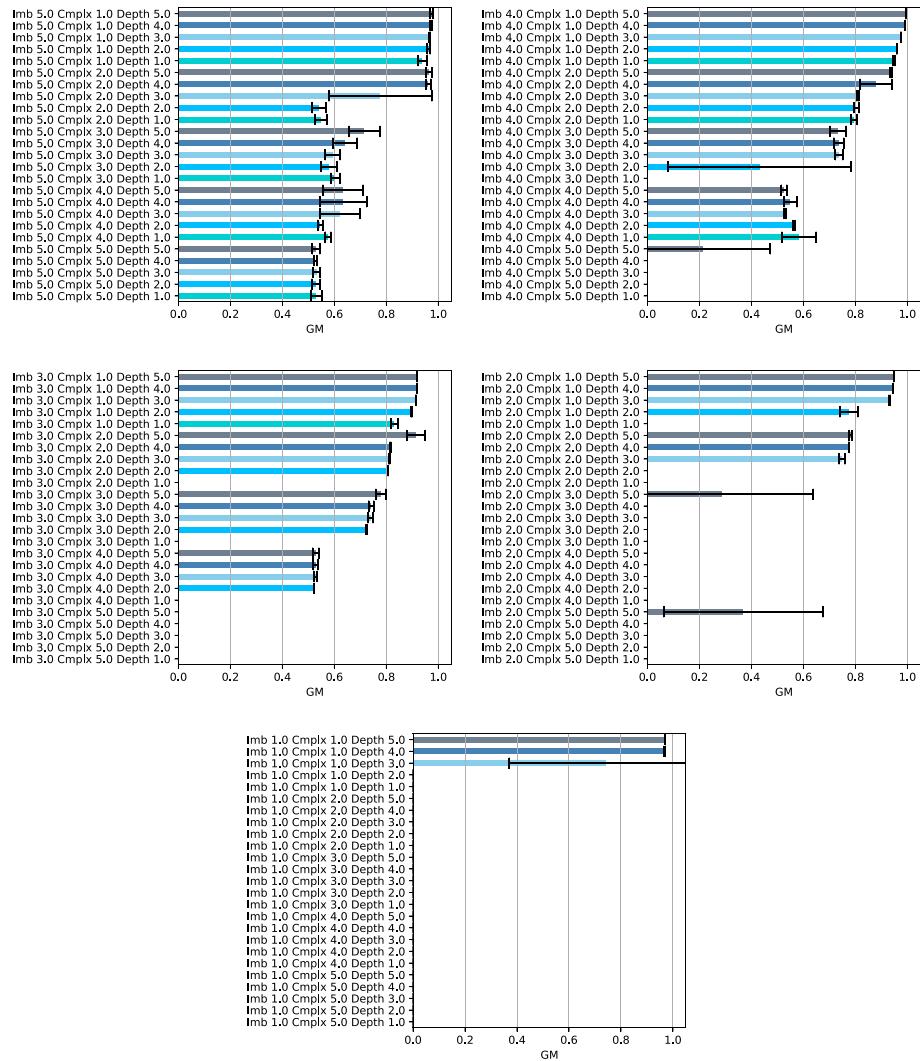


Fig. 18 The effect of depth on class imbalance levels, complexity in small datasets in the Backbone Domain. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

E Additional results on depth for convolutional neural networks

This part of the appendix provides additional results pertaining to the effect of Depth on Convolutional Neural Networks in the cases of class imbalances, concept complexity and data scarcity. These results were not included in the main body of the paper to improve the presentation of the results. However, we felt that it was important to include them for the interested reader. Figure 21 shows the results obtained on all depths for the Backbone Domain of small size ($s = 1$). Figures 22 and 23 show the results obtained on the image domains, namely MNIST-Fashion and CIFAR-10. These results are discussed in the main body of the paper.

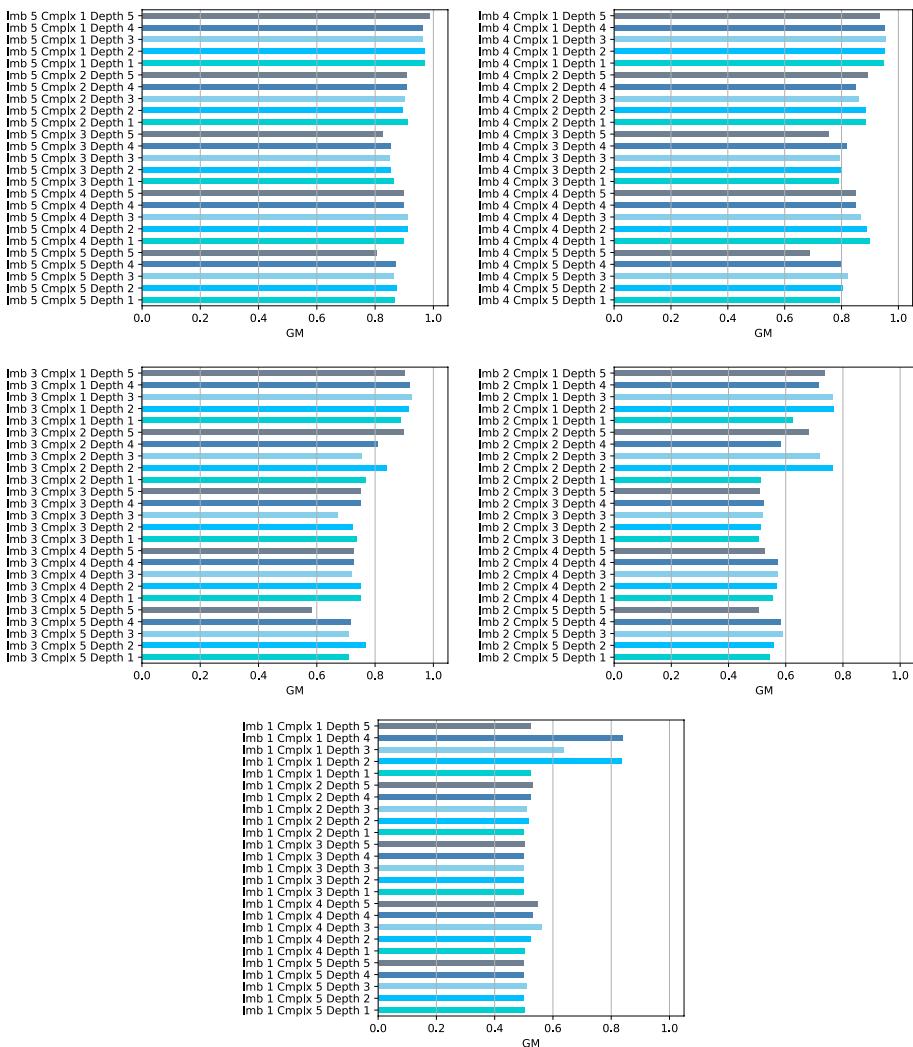


Fig. 19 The effect of depth on class imbalance levels, complexity in small datasets in the newsgroup domain. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

F Additional results for regularization on multi layer perceptrons

This part of the appendix provides additional results pertaining to the effect of Regularization on Multi Layer Perceptrons in the cases of class imbalances, concept complexity and data scarcity. These results were not included in the main body of the paper to improve the presentation of the results. However, we felt that it was important to include them for the interested reader. Figure 24 shows the results obtained on all depths for the large Backbone

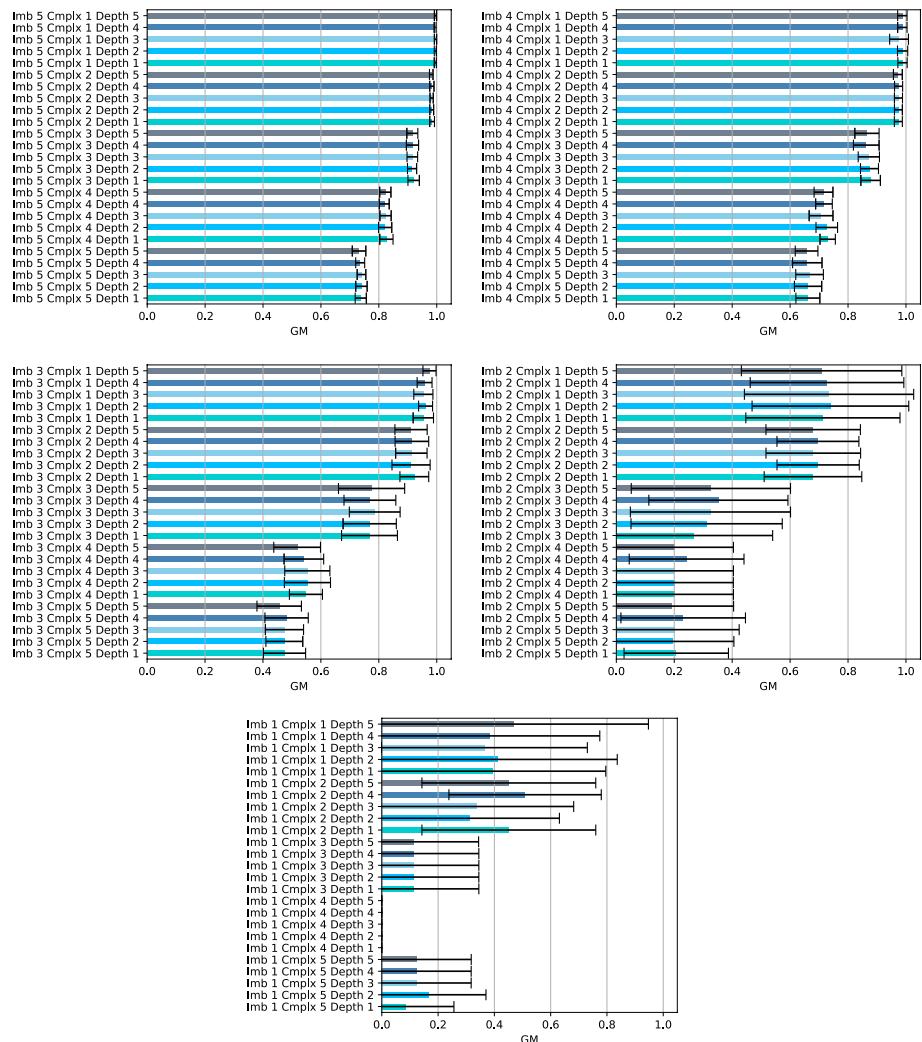


Fig. 20 The effect of depth on class imbalance levels, complexity in the Job Classification domain. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

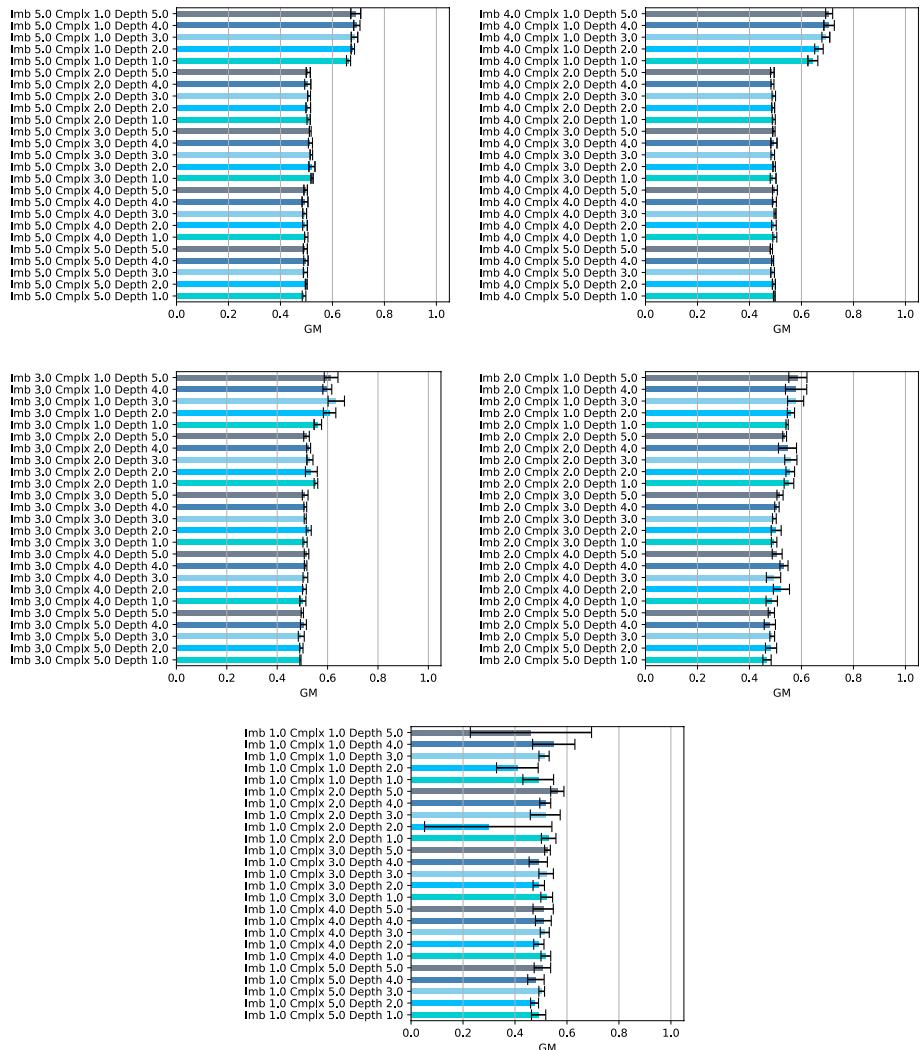


Fig. 21 The effect of depth on class imbalance levels, complexity in small datasets in the Shapes domain. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

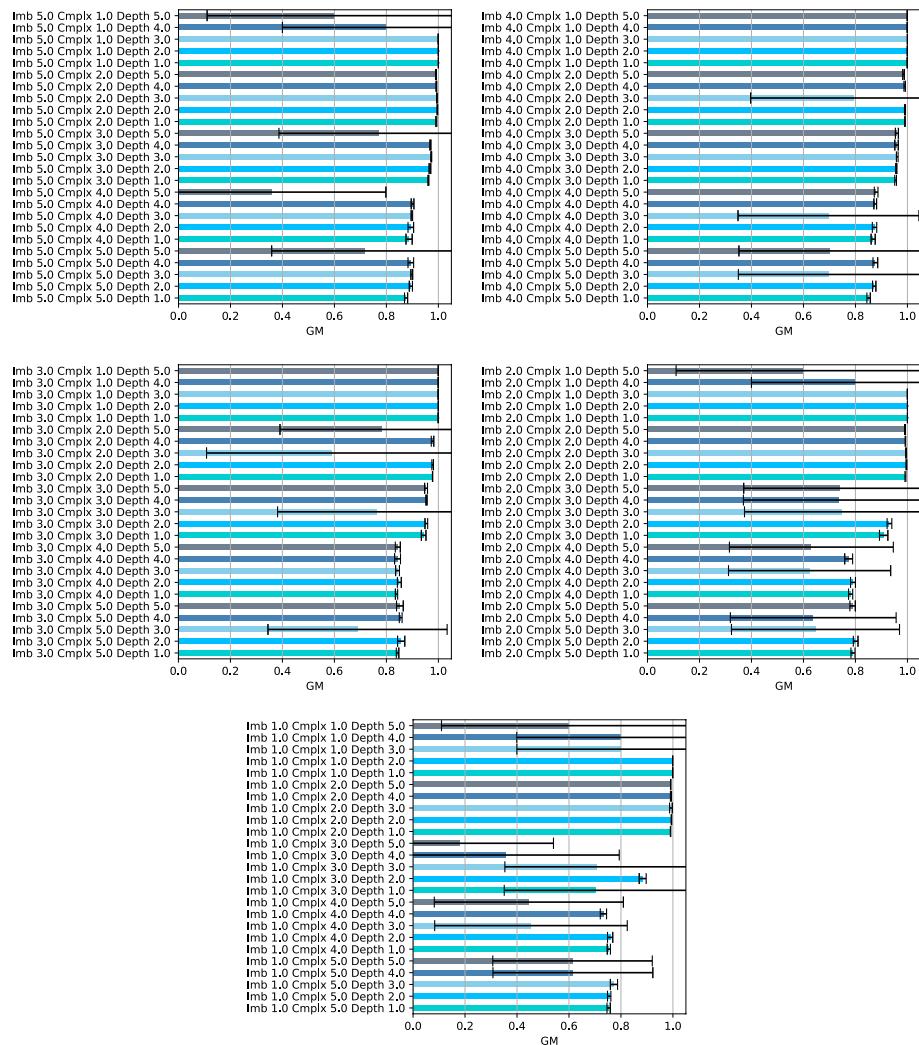


Fig. 22 The effect of depth on class imbalance levels, complexity in MNIST Fashion datasets. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

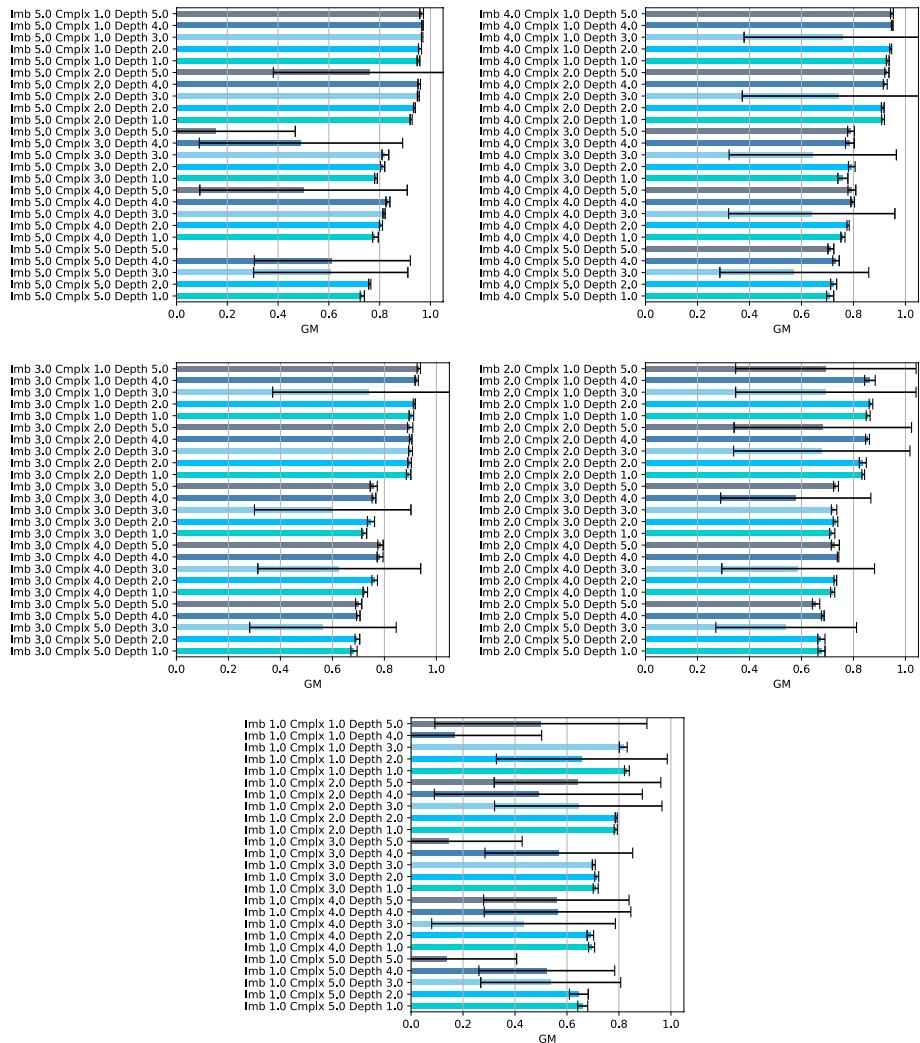


Fig. 23 The effect of depth on class imbalance levels, complexity in CIFAR-10 datasets. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

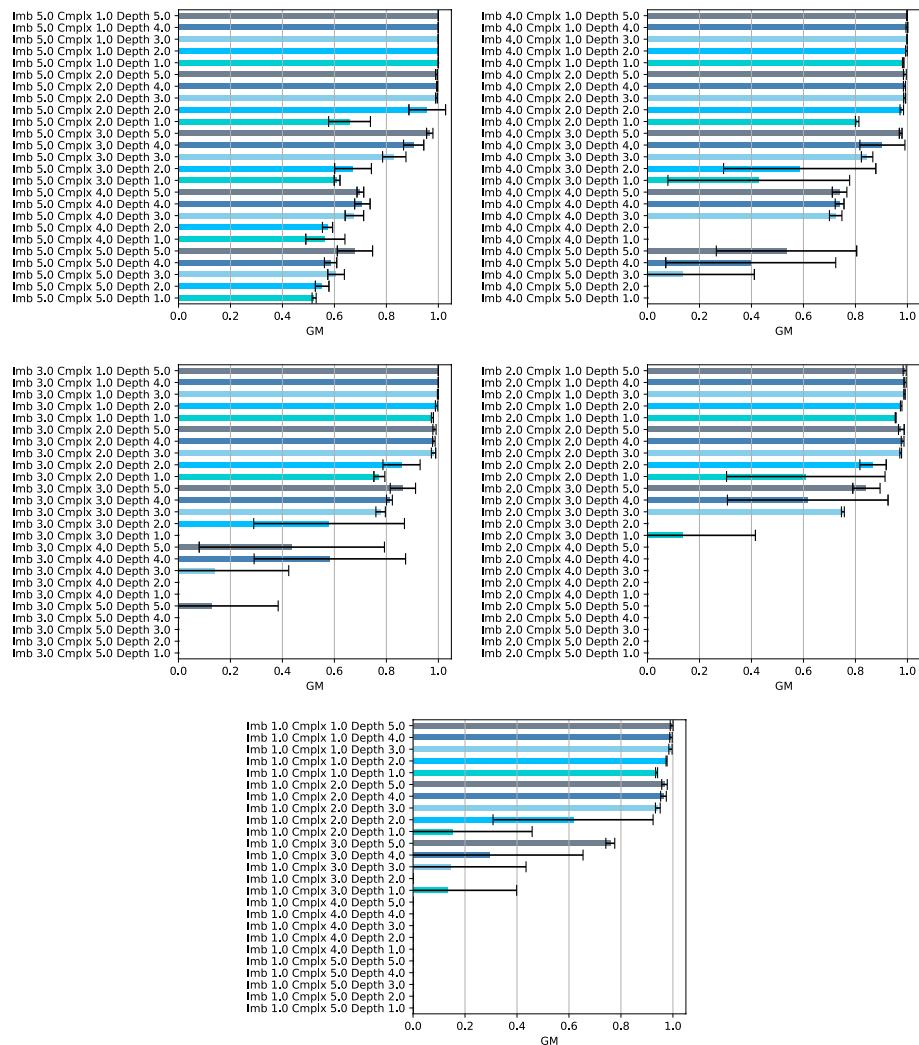


Fig. 24 The effect of regularization on the backbone experiments. Size = 5. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

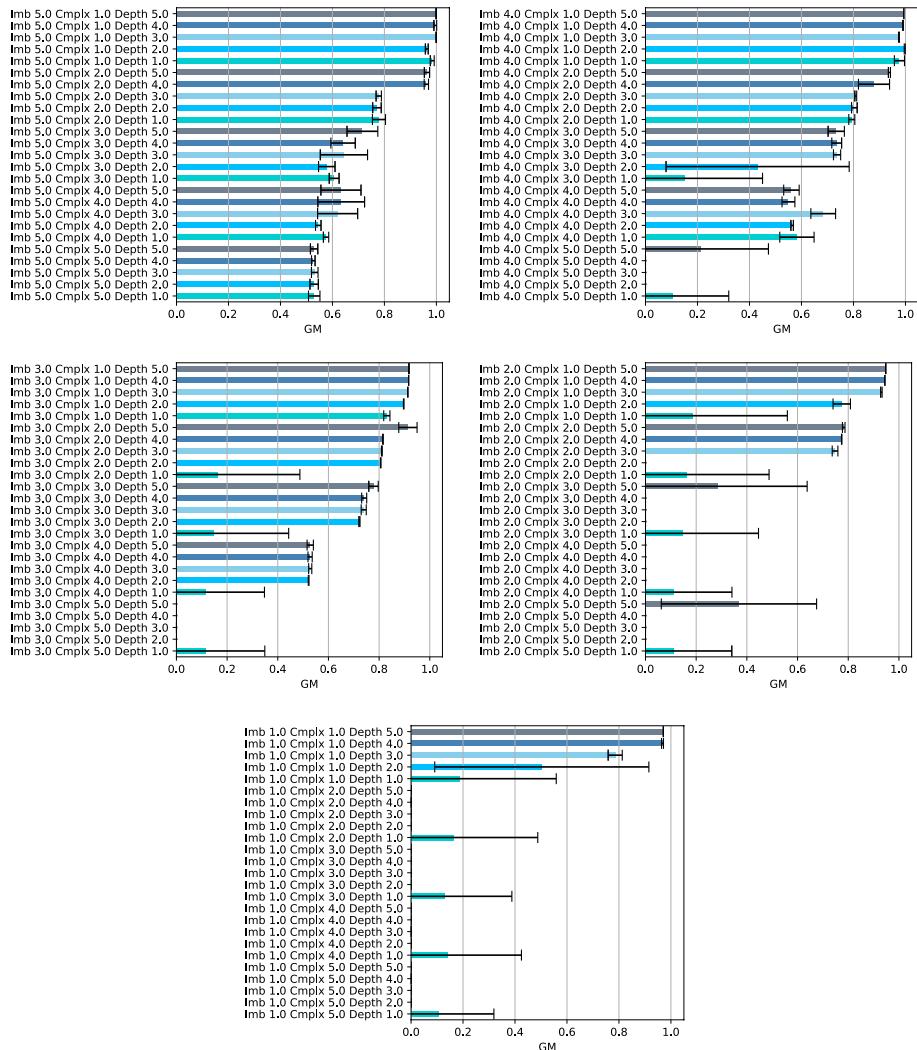


Fig. 25 The effect of Regularization on the backbone experiments. Size = 1. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

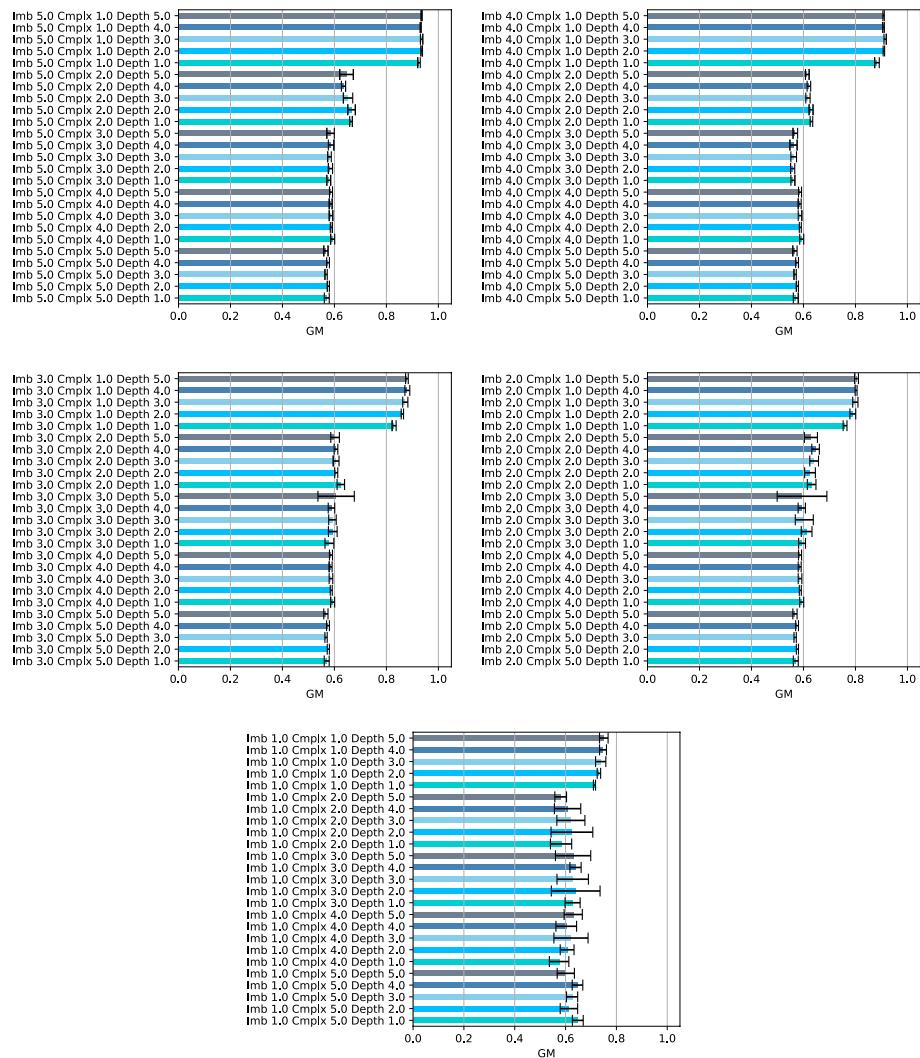


Fig. 26 The effect of Regularization on the Shape domain. Size = 5. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

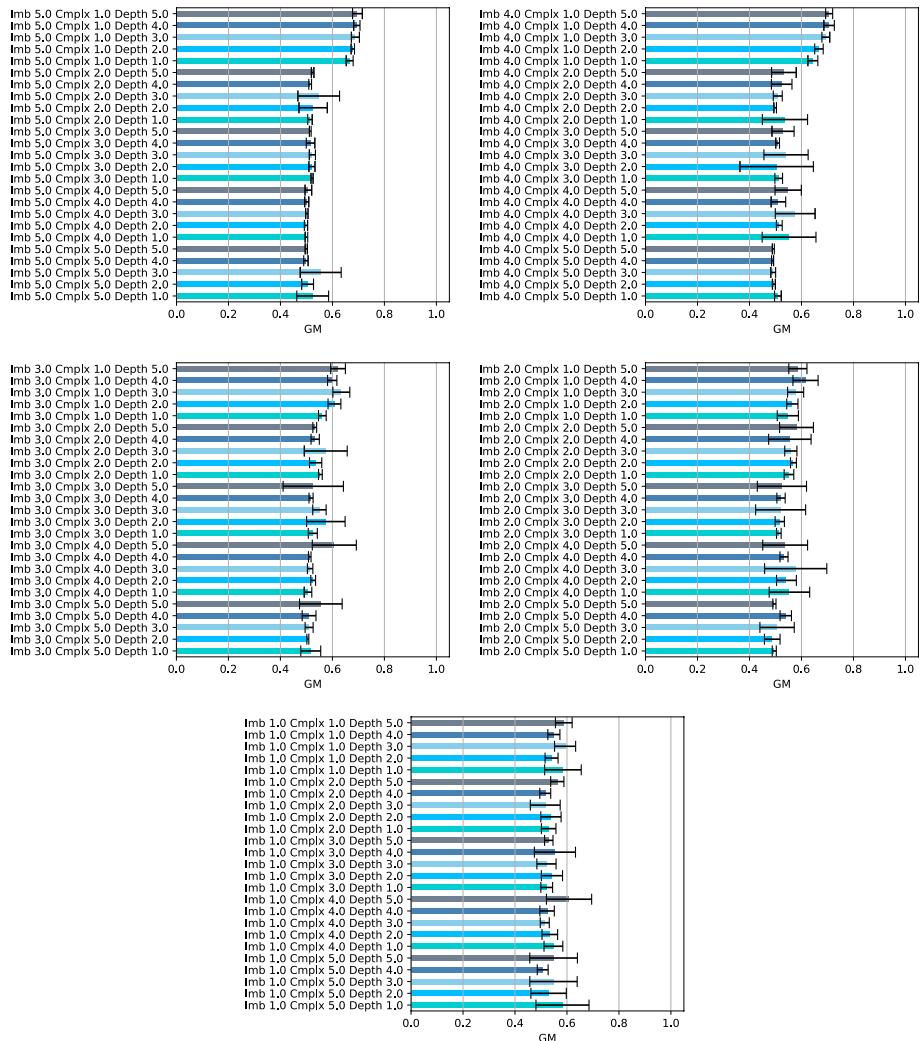


Fig. 27 The effect of Regularization on the Shape domain. Size = 1. Least imbalance: level 5 (leftmost); Most imbalance: level 1 (rightmost). Each cluster of five bars represent a complexity level, going from low to high complexity. Within each cluster, each bar represents a depth level going from deeper on top to shallower at the bottom of the cluster

Domain (size $s = 5$) while Fig. 25 shows the results obtained on all depths for the small Backbone Domain (size $s = 1$). These results are discussed in the main body of the paper.

G Additional results for regularization on convolutional neural networks

This part of the appendix provides additional results pertaining to the effect of Regularization on Convolutional Neural Networks in the cases of class imbalances, concept complexity and data scarcity. These results were not included in the main body of the paper to improve the presentation of the results. However, we felt that it was important to include them for the interested reader. Figure 26 shows the results obtained on all depths for the large Shape Domain (size $s = 5$) while Fig. 27 shows the results obtained on all depths for the small Shape Domain (size $s = 1$). These results are discussed in the main body of the paper.

Acknowledgements We would like to thank Julia Chen and Becca Jeffries for their contributions on the shape and text classification experiments.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Kushankur Ghosh, Colin Bellinger, Roberto Corizzo and Paula Branco. The first draft of the manuscript was written by Nathalie Japkowicz, edited by Bartosz Krawczyk and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The authors did not receive support from any organization for the submitted work.

Data availability The results on other performance metrics as well as stratified cross-validated results, that were not included in the manuscript, are available upon request. The code for the experiments can be accessed here: <https://github.com/cbellinger27/deep-imbalance-analysis>.

Declarations

Conflicts of interest The authors have no relevant financial or non-financial interests to disclose.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethics approval Not applicable.

References

- Alhenaki, L., & Hosny, M. (2019). A genetic-frog leaping algorithm for large dataset document clustering. In *2019 IEEE/ACS 16th international conference on computer systems and applications (AICCSA)*, IEEE, pp 1–4.
- Anand, R., Mehrotra, K. G., Mohan, C. K., & Ranka, S. (1993). An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6), 962–969.
- Barella, V. H., Garcia, L. P., de Souto, M. C., Lorena, A. C., & de Carvalho, A. C. (2021). Assessing the data complexity of imbalanced datasets. *Information Sciences*, 553, 83–109.

- Bellinger, C., Corizzo, R., & Japkowicz, N. (2021). Calibrated resampling for imbalanced and long-tails in deep learning. In *International conference on discovery science*, Springer, pp. 242–252.
- Benitez-Quiroz, C. F., Srinivasan, R., Feng, Q., Wang, Y., & Martinez, A. M. (2017). Emotionet challenge: Recognition of facial expressions of emotion in the wild. arXiv preprint [arXiv:1703.01210](https://arxiv.org/abs/1703.01210)
- Bhatia, S., & Dahyot, R. (2019). Using wgan for improving imbalanced classification performance. In *CEUR Workshop Proceedings, CEUR*, 2563, 365–375.
- Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2), 1–50.
- Bria, A., Marrocco, C., & Tortorella, F. (2020). Addressing class imbalance in deep learning for small lesion detection on medical images. *Computers in Biology and Medicine*, 120, 103735.
- Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249–259.
- Cano, J. R. (2013). Analysis of data complexity measures for classification. *Expert Systems with Applications*, 40(12), 4820–4831.
- Cao, K., Wei, C., Gaidon, A., Arechiga, N., & Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in neural information processing systems*, 32.
- Chou, H. P., Chang, S. C., Pan, J. Y., Wei, W., & Juan, D. C. (2020). Remix: Rebalanced mixup. In *European conference on computer vision*, Springer, pp. 95–110.
- Cohen, G., Sapiro, G., & Giryes, R. (2018). DNN or k-NN: That is the generalize vs. memorize question. arXiv preprint [arXiv:1805.06822](https://arxiv.org/abs/1805.06822).
- Cui, Y., Jia, M., Lin, T. Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 9268–9277.
- Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15. <https://doi.org/10.1109/TNNLS.2021.3136503>.
- Ding, W., Huang, D. Y., Chen, Z., Yu, X., & Lin, W. (2017). Facial action recognition using very deep networks for highly imbalanced class distribution. In *2017 Asia-Pacific signal and information processing association annual summit and conference (APSIPA ASC)*, IEEE, pp. 1368–1372.
- Dong, Q., Gong, S., & Zhu, X. (2017). Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE international conference on computer vision*, pp. 1851–1860.
- Drummond, C., & Holte, R. (2003). Class imbalance and cost sensitivity: Why undersampling beats oversampling. In *ICML-KDD 2003 workshop: Learning from imbalanced datasets*, vol. 3.
- Du, J., Zhou, Y., Liu, P., Vong, C. M., & Wang, T. (2021). Parameter-free loss for class-imbalanced deep learning in image classification. *IEEE Transactions on Neural Networks and Learning Systems*.
- Dudjak, M., & Martinović, G. (2021). An empirical study of data intrinsic characteristics that make learning from imbalanced data difficult. *Expert Systems with Applications*, p. 115297.
- El Korchi, A., & Ghanou, Y. (2020). 2D geometric shapes dataset—for machine learning and pattern recognition. *Data in Brief*, 32.
- Fernández, A., García, S., Galarraga, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*. Springer. <https://doi.org/10.1007/978-3-319-98074-4>.
- Fuqua, D., & Razzaghi, T. (2020). A cost-sensitive convolution neural network learning for control chart pattern recognition. *Expert Systems with Applications*, 150, 113275.
- Geng, Y., & Luo, X. (2018). Cost-sensitive convolution based neural networks for imbalanced time-series classification. arXiv preprint [arXiv:1801.04396](https://arxiv.org/abs/1801.04396).
- Ghosh, K., Bellinger, C., Corizzo, R., Krawczyk, B., & Japkowicz, N. (2021). On the combined effect of class imbalance and concept complexity in deep learning. In *Proceedings of the IEEE big data conference*, pp. 4859–4868.
- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings*, pp. 249–256.
- Guzy, F., Wozniak, M., & Krawczyk, B. (2021). Evaluating and explaining generative adversarial networks for continual learning under concept drift. In *2021 International conference on data mining, ICDM 2021 - workshops*, Auckland, New Zealand
- Hensman, P., & Masko, D. (2015). *The impact of imbalanced training data for convolutional neural networks*. KTH Royal Institute of Technology: Degree Project in Computer Science.
- Herland, M., Khoshgoftaar, T. M., & Bauder, R. A. (2018). Big data fraud detection using multiple medicare data sources. *Journal of Big Data*, 5(1), 1–21.

- Ho, T. K., & Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3), 289–300.
- Hu, J., Zhang, H., Liu, Y., Sutcliffe, R., & Feng, J. (2021). Bbw: a batch balance wrapper for training deep neural networks on extremely imbalanced datasets with few minority samples. *Applied Intelligence*, pp. 1–16.
- Huang, C., Li, Y., Loy, C. C., & Tang, X. (2016). Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384.
- Huang, C., Li, Y., Loy, C. C., & Tang, X. (2019). Deep imbalanced learning for face recognition and attribute prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(11), 2781–2794.
- Huang, C., Wu, X., Zhang, X., Lin, S., & Chawla, N. V. (2019b). Deep prototypical networks for imbalanced time series classification under data scarcity. In *Proceedings of the 28th ACM international conference on information and knowledge management, CIKM 2019*, Beijing, China, November 3–7, 2019, ACM, pp. 2141–2144.
- Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5), 429–449.
- Johnson, J. M., & Khoshgoftaar, T. M. (2020). The effects of data sampling with deep learning and highly imbalanced big data. *Information Systems Frontiers*, 22(5), 1113–1131.
- Johnson, J. M., & Khoshgoftaar, T. M. (2021). Thresholding strategies for deep learning with highly imbalanced big data. In *Deep learning applications*, vol. 2, Springer, pp. 199–227.
- Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., & Kalantidis, Y. (2019). Decoupling representation and classifier for long-tailed recognition. arXiv preprint [arXiv:1910.09217](https://arxiv.org/abs/1910.09217).
- Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3573–3587.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
- Korycki, L., & Krawczyk, B. (2021). Class-incremental experience replay for continual learning under concept drift. In *IEEE conference on computer vision and pattern recognition workshops, CVPR workshops 2021, virtual, June 19–25, 2021, Computer Vision Foundation/IEEE*, pp. 3649–3658.
- Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5(4), 221–232.
- Krizhevsky, A., & Hinton, G., et al. (2009) Learning multiple layers of features from tiny images.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Lee, H., Park, M., & Kim, J. (2016). Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, IEEE, pp. 3713–3717.
- Li, B., Liu, Y., & Wang, X. (2019). Gradient harmonized single-stage detector. *Proceedings of the AAAI conference on artificial intelligence*, 33, 8577–8584.
- Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint [arXiv:1312.4400](https://arxiv.org/abs/1312.4400).
- Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.
- Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2019). On the variance of the adaptive learning rate and beyond. arXiv preprint [arXiv:1908.03265](https://arxiv.org/abs/1908.03265).
- Llamas, J., Lerones, P. M., Medina, R., Zalama, E., Gómez-García-Bermejo, J. (2017). Classification of architectural heritage images using deep learning techniques. *Applied Sciences*, 7(10):992.
- López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, 250, 113–141.
- Luengo, J., Fernández, A., García, S., & Herrera, F. (2011). Addressing data complexity for imbalanced data sets: Analysis of smote-based oversampling and evolutionary undersampling. *Soft Computing*, 15(10), 1909–1936.

- Luo, L., Xiong, Y., Liu, Y., & Sun, X. (2019). Adaptive gradient methods with dynamic bound of learning rate. arXiv preprint [arXiv:1902.09843](https://arxiv.org/abs/1902.09843).
- Lydia, A., & Francis, S. (2019). Adagrad: An optimizer for stochastic gradient descent. *International Journal of Computer and Information Sciences*, 6(5).
- Lyon, R. J., Stappers, B., Cooper, S., Brooke, J. M., & Knowles, J. D. (2016). Fifty years of pulsar candidate selection: From simple filters to a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society*, 459(1), 1104–1123.
- Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., & Malossi, C. (2018). Bagan: Data augmentation with balancing gan. arXiv preprint [arXiv:1803.09655](https://arxiv.org/abs/1803.09655).
- Mullick, S. S., Datta, S., & Das, S. (2019). Generative adversarial minority oversampling. In *2019 IEEE/CVF international conference on computer vision, ICCV 2019*, Seoul, Korea (South), October 27–November 2, 2019, IEEE, pp. 1695–1704.
- Murphrey, Y. L., Guo, H., & Feldkamp, L. A. (2004). Neural learning from unbalanced data. *Applied Intelligence*, 21(2), 117–128.
- Nazari, E., & Branco, P. (2021). On oversampling via generative adversarial networks under different data difficulty factors. In *Third international workshop on learning with imbalanced domains: Theory and applications*, pp. 76–89.
- Ng, W. W., Liu, Z., Zhang, J., & Pedrycz, W. (2021). Maximizing minority accuracy for imbalanced pattern classification problems using cost-sensitive localized generalization error model. *Applied Soft Computing*, 104, 107178.
- Orenstein, E. C., Bejbom, O., Peacock, E. E., & Sosik, H. M. (2015). Whoi-plankton-a large scale fine grained visual recognition benchmark dataset for plankton classification. arXiv preprint [arXiv:1510.00745](https://arxiv.org/abs/1510.00745).
- Peng, D., Gu, T., Hu, X., & Liu, C. (2021). Addressing the multi-label imbalance for neural networks: An approach based on stratified mini-batches. *Neurocomputing*, 435, 91–102.
- Pouyanfar, S., Tao, Y., Mohan, A., Tian, H., Kaseb, A. S., Gauen, K., Dailey, R., Aghajanzadeh, S., Lu, Y. H., & Chen, S. C., et al. (2018). Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, IEEE, pp. 112–117.
- Prati, R. C., Batista, G. E., & Monard, M. C. (2004). Class imbalances versus class overlapping: An analysis of a learning system behavior. In *Mexican international conference on artificial intelligence*, Springer, pp. 312–321.
- Pulgar, F. J., Rivera, A. J., Charte, F., & del Jesus, M. J. (2017). On the impact of imbalanced data in convolutional neural networks performance. In *International conference on hybrid artificial intelligence systems*, Springer, pp. 220–232.
- Raj, V., Magg, S., & Wermter, S. (2016). Towards effective classification of imbalanced data with convolutional neural networks. In *IAPR workshop on artificial neural networks in pattern recognition*, Springer, pp. 150–162.
- Ren, M., Zeng, W., Yang, B., & Urtasun, R. (2018). Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343.
- Ross, T. Y., & Dollár, G. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2980–2988.
- Sáez, J. A., Krawczyk, B., & Wozniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57, 164–178.
- Sampath, V., Maurtua, I., Martín, J. J. A., & Gutierrez, A. (2021). A survey on generative adversarial networks for imbalance problems in computer vision tasks. *Journal of Big Data*, 8(1), 1–59.
- Santos, M. S., Abreu, P. H., Japkowicz, N., Fernández, A., Soares, C., Wilk, S., & Santos, J. (2022). On the joint-effect of class imbalance and overlap: A critical review. *Artificial Intelligence Review*, pp. 1–69.
- Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2008a). A comparative study of data sampling and cost sensitive learning. In *2008 IEEE international conference on data mining workshops*, IEEE, pp. 46–52.
- Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2008b). Resampling or reweighting: A comparison of boosting implementations. In *2008 20th IEEE international conference on tools with artificial intelligence*, IEEE, 1:445–451.
- Sellami, A., & Hwang, H. (2019). A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. *Expert Systems with Applications*, 122, 75–84.
- Sikora, M., et al. (2010). Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines. *Archives of Mining Sciences*, 55(1), 91–114.

- Simon, M., Rodner, E., & Denzler, J. (2016). Imagenet pre-trained models with batch normalization. arXiv preprint [arXiv:1612.01452](https://arxiv.org/abs/1612.01452).
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- Sleeman, W. C., & Krawczyk, B. (2021). Multi-class imbalanced big data classification on spark. *Knowledge-Based Systems*, 212, 106598. <https://doi.org/10.1016/j.knosys.2020.106598>
- Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, IEEE, pp. 464–472.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint [arXiv:1412.6806](https://arxiv.org/abs/1412.6806).
- Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32, 323–332.
- Telgarsky, M. (2016). Benefits of depth in neural networks. In *Conference on learning theory*, pp. 1517–1539.
- Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. (2017). Wasserstein auto-encoders. arXiv preprint [arXiv:1711.01558](https://arxiv.org/abs/1711.01558).
- Valova, I., Harris, C., Mai, T., & Gueorguieva, N. (2020). Optimization of convolutional neural networks for imbalanced set classification. *Procedia Computer Science*, 176, 660–669.
- Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2011). Class imbalance, redux. In *2011 IEEE 11th international conference on data mining*, IEEE, pp. 754–763.
- Wang, H., Cui, Z., Chen, Y., Avidan, M., Abdallah, A. B., & Kronzer, A. (2018). Predicting hospital readmission via cost-sensitive deep learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(6), 1968–1978.
- Wang, J., & Jean, J. (1993). Resolving multifont character confusion with neural networks. *Pattern Recognition*, 26(1), 175–187.
- Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., & Kennedy, P. J. (2016). Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, IEEE, pp. 4368–4374.
- Wang, X., Lian, L., Miao, Z., Liu, Z., & Yu, S.X. (2020a). Long-tailed recognition by routing diverse distribution-aware experts. arXiv preprint [arXiv:2010.01809](https://arxiv.org/abs/2010.01809).
- Wang, X., Lyu, Y., & Jing, L. (2020b). Deep generative model for robust imbalance classification. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020*, Seattle, WA, USA, June 13–19, 2020, IEEE, pp. 14112–14121.
- Weiss, G. M., McCarthy, K., Zabar, B. (2007). Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *Dmin*, 7(35–41):24.
- Ya-Guan, Q., Jun, M., Xi-Min, Z., Jun, P., Wu-Jie, Z., Shu-Hui, W., Ben-Sheng, Y., & Jing-Sheng, L. (2020). Emsgd: An improved learning algorithm of neural networks with imbalanced data. *IEEE Access*, 8, 64086–64098.
- Ye, H. J., Zhan, D. C., & Chao, W. L. (2021). Procrustean training for imbalanced deep learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 92–102.
- Zaheer, M., Reddi, S., Sachan, D., Kale, S., & Kumar, S. (2018). Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31.
- Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. arXiv preprint [arXiv:1212.5701](https://arxiv.org/abs/1212.5701).
- Zheng, Y., Liu, Q., Chen, E., Ge, Y., & Zhao, J. L. (2014). Time series classification using multi-channels deep convolutional neural networks. In *International conference on web-age information management*. Springer, pp. 298–310.
- Zhou, B., Cui, Q., Wei, X. S., & Chen, Z. M. (2020). BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9719–9728.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Kushankur Ghosh¹ · Colin Bellinger² · Roberto Corizzo³ · Paula Branco⁴  ·
Bartosz Krawczyk⁵ · Nathalie Japkowicz³

Kushankur Ghosh
kushanku@ualberta.ca

Colin Bellinger
Colin.Bellinger@nrc-cnrc.gc.ca

Roberto Corizzo
rcorizzo@american.edu

Bartosz Krawczyk
bkrawczyk@vcu.edu

Nathalie Japkowicz
japkowic@american.edu

¹ Department of Computing Science, University of Alberta, Edmonton, Canada

² National Research Council of Canada, Ottawa, Canada

³ Department of Computer Science, American University, Washington, DC, USA

⁴ School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada

⁵ Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA