User Modeling and User-Adapted Interaction
https://doi.org/10.1007/s11257-023-09381-y

Solving the imbalanced data issue: automatic urgency detection for instructor assistance in MOOC discussion forums

Laila Alrajhi[1,2] · Ahmed Alamri[3] · Filipe Dwan Pereira[4] · Alexandra I. Cristea[1] · Elaine H. T. Oliveira[5]

Received: 13 March 2023 / Accepted in revised form: 9 August 2023

Abstract

In MOOCs, identifying urgent comments on discussion forums is an ongoing challenge. Whilst urgent comments require immediate reactions from instructors, to improve interaction with their learners, and potentially reducing drop-out rates—the task is difficult, as truly urgent comments are rare. From a data analytics perspective, this represents a *highly unbalanced (sparse) dataset*. Here, we aim to *automate the urgent comments identification process, based on fine-grained learner modelling*—to be used for automatic recommendations to instructors. To showcase and compare these models, we apply them to the *first gold standard dataset for* ***U****rgent i****N****structor* ***I****n****TE****rvention (UNITE)*, which we created by labelling FutureLearn MOOC data. We implement both benchmark shallow classifiers and deep learning. Importantly, we not only *compare, for the first time for the unbalanced problem, several data balancing techniques*, comprising text augmentation, text augmentation with undersampling, and undersampling, but also *propose several new pipelines for combining different augmenters for text augmentation*. Results show that models with undersampling can predict most urgent cases; and 3X *augmentation* + *undersampling* usually attains the best performance. We additionally validate the best models via a generic benchmark dataset (Stanford). As a case study, we showcase how the naïve Bayes with count vector can adaptively support instructors in answering learner questions/comments, potentially saving time or increasing efficiency in supporting learners. Finally, we show that the errors from the classifier mirrors the disagreements between annotators. Thus, our proposed algorithms perform at least as well as a 'super-diligent' human instructor (with the time to consider all comments).

Keywords MOOCs · Urgent comments · Natural language processing · Machine learning · Imbalanced data · Text augmentation · Undersampling · Adaptive models · Error analysis

Extended author information available on the last page of the article

Published online: 01 December 2023

1 Introduction

Massive Open Online Course (MOOC) platforms continue to grow dramatically; in recent years, Coursera, edX and FutureLearn have emerged as popular platforms (Joseph 2020). They support a wide variety of efficiently delivered courses with easy access, which open numerous learning opportunities to anyone wanting to learn a specific topic or obtain new information. As the popularity of MOOCs continues to grow, ever more people with a broad diversity of background knowledge and goals are enrolling on these platforms from around the world. For over a decade now, many research communities have contributed to the development of these platforms and proposed specific solutions to the challenges and barriers they face, such as learner engagement (Anderson et al. 2014), learner motivation (Durksen et al. 2016) and learner performance (Jiang et al. 2014). One way to lift some of these barriers is via *accurate and on-time instructor intervention on MOOC discussion forums* (Alrajhi et al. 2021).

Discussion forums enable online learners to express their ideas, ask questions, and seek help (Crossley et al. 2015). In addition, they create social connections and facilitate communication among learners and instructors (Stump et al. 2013). In this context, instructors play an important role in monitoring comments, especially to provide the help and support learners need: an accurate on-time intervention may make the difference between a learner continuing on the course or dropping out; indeed, (Alrajhi et al. 2021) showed that learners are less likely to finish the course (about 13%) if they frequently make comments that require intervention. However, continuously monitoring such huge numbers of comments is a time-consuming and sometimes overwhelming task for instructors: hundreds or even thousands of comments are posted during courses, sometimes for each course step, and identifying those which require urgent intervention can be almost impossible. This is exacerbated by the high ratio of learners to instructors (Almatrafi and Johri 2018).

As MOOCs generate huge amounts of textual data, another way of addressing this issue is via Natural Language Processing (NLP). The work presented in this paper uses NLP to help instructors to address urgent comments and enable them to decide when to react, by creating an automatic text-classification model.

Another core problem in this area is the intrinsically imbalanced nature of the data; such datasets are characterised by a highly skewed class distribution due to the (naturally) small number of 'urgent comment' instances. In text classification tasks, performance often depends on the quality of the data (Wei and Zou 2019). Therefore, to tackle the imbalanced data problem and improve the size and quality of the training data, we manipulate the dataset by: *text augmentation*, *text augmentation with undersampling*, and *undersampling*.

To illustrate the usage of the fine-grained learner models in adaptive support for instructor intervention, we describe an adaptation case where instructors can decrease their workload by using one of our models. We also showcase an expanded model that uses more extensive learner knowledge (based on the number of comments per learner), to discuss how such adaptation models can be further expanded.

1.1 Contributions

The main contributions of our research are, to the best of our knowledge:

- Creating the first learner, instructor and adaptation models to support instructors to deal with urgent comments in MOOCs.
- For the first time in the literature, applying data balancing techniques for shallow and deep machine learning to identify instances when urgent instructor intervention is required on MOOCs. These techniques include text augmentation, text augmentation with undersampling, and undersampling to overcome the imbalanced data problem and improve performance. This is achieved by 'forcing' the algorithm to increase the weight of the minority class.
- Creating the first *gold standard corpus MOOC* ***U****rgent i****N****structor* ***InTE****rvention (UNITE)* for instructor intervention in MOOC environments (the FutureLearn platform), which has been annotated by carefully selected experts in the field. This will be made available (after ethical cleansing) to the research community.
- Proposing several new pipelines (3X and 9X) to generate more data for text augmentation by incorporating different NLP augmenters and providing a range of approaches.
- Showcasing the challenges and difficulties involved in instructor-intervention decisions in MOOC environments, by manually inspecting and analysing the (relatively small) set of errors generated by the best classifier, along with the best data balancing and text augmentation solutions.

2 Literature review

Today, the instructor intervention problem is one of the most challenging in MOOC environments. Separately, a related, even less explored area of research has emerged, identifying the difficult area of urgent posts detection (Almatrafi et al. 2018; Guo et al. 2019; Alrajhi et al. 2020; Khodeir 2021). However, an obvious omission is that for urgent posts, imbalanced data are a characteristic of the data itself (as there are less urgent comments than non-urgent, normally). This fact has been overlooked in urgent post detection. The closest research to this (Almatrafi et al. 2018; Khodeir 2021) considered some standard techniques: splitting the data, training the model, and selecting the evaluation metrics, but not dealing with improve data imbalance. In addition, while available intervention models for urgent comments concentrated on classifying posts, they did not pay any attention to the behaviours of learners or designed adaptive instructor intervention models based on learner (or instructor) models. Therefore, this section reviews the literature areas closest to our proposal: (1) the important area of instructor intervention in MOOCs, focusing on urgent posts, (2) the area of text augmentation, specifically for balancing data, and (3) adaptive models in MOOCs.

2.1 Instructor intervention in MOOC forums

In 2012, the first MOOC-like discussion forums were developed and immediately aroused researchers' interest. According to (Almatrafi and Johri 2018), 234 researchers inspected discussion forums from 2013 to 2018. Only as recently as 2014, (Chaturvedi et al. 2014) first investigated the intervention problem, by building numerous models to predict which forum discussion thread instructors should intervene on. They utilised course information, forum structure, and post content; importantly, they also considered information on whether the next post to be written was by an instructor, hence enlisting characteristics of real instructor behaviour. Similarly, (Chandrasekaran et al. 2015b) built a classifier that considered prior knowledge of the forum type. These researchers used the Coursera platform and trained on historical instructor interventions. This approach, we argue, is inadequate, since (historical) instructor intervention likely resulted from a subjective decision to offer support. Moreover, it is arguably based on decisions on a *subset* of posts, because instructors may not have had sufficient time to read *all* the posts related to a particular course, to decide which were urgent (Chandrasekaran et al. 2015b).

The first research to use the Stanford MOOC Post dataset (Bakharia 2016) proposed a generalizable transfer-learning-based model to identify urgency as one of three forum-post classifications (*confusion, urgency,* and *sentiment),* by applying a cross-domain approach. Whilst the model failed to obtain adequate results, the author recommended transfer-learning as worthy of further research. Wei et al. (2017) followed the same cross-domain technique but applied a deep neural network element; this increased performance.

Almatrafi et al. (2018) also utilised the Stanford MOOC Post dataset to classify urgent posts, by training different shallow classifiers and proposing the best features for them. Sun et al. (2019) used instead deep learning, via improved recurrent convolutional neural networks (RCNN), achieving higher performance in identifying urgent posts compared to other models (naïve Bayes, SVM (RBF), random forest, CNN, RNN, LSTM, GRU, and RCNN). Another work by (Guo et al. 2019) proposed a hybrid neural network based on the attention mechanism to recognise urgent posts. With a similar goal, Alrajhi et al. 2020 used a multidimensional model to determine urgent posts requiring intervention, comparing two different models: (i) text-only, and (ii) text and numerical data. The findings highlight that the combined, multi-dimensional-features model is more effective than the text-only (NLP) analysis. Clavié and Gal (2019) created EduBERT, a contextualised word-embedding technique: it represents the current state-of-the-art performance on classifying urgent posts using EduDistilBERT (0.835 in Recall for the minority class). Khodeir (2021) built an urgency classification model, which is based on a fine-tuned BERT as an embedding layer feeding it into a multi-layer bi-directional GRU, and she reported their results based on three groups with (0.815, 0.847 and 0.831 in Recall), which is close to the state-of-the-art.

Another work that used the Stanford MOOC Post dataset for the intervention task (Capuano and Caballé, 2019) proposed a text categorisation tool for a multi-attribute categorisation of MOOC forum posts; one of these attributes is a level of urgency, with preliminary results to use for intervention, or as input for conversational agents (chatbots). Their follow-up study (Capuano et al. 2021) is an improvement of their tool, using attention-based hierarchical recurrent neural networks. However, their work classified urgency into three categories (low, medium and high) and reported an average recall (R), instead of the per class R. In addition, (Rossi et al. 2021) detected which type of pedagogical intervention is required, based on a conversational agent, using an ontology and a set of semantic rules.

Another study conducted by (Toti et al. 2020) built on the approach in (Capuano and Caballé, 2019); created a methodology to detect engagement in e-learning platforms and to help instructors with their timeliness of their interventions, based on different aspects, one of these being urgency, detected as a classification task; however, their work lacked the implementation.

The vast majority of recently published research on urgent post classification uses the Stanford MOOC Post dataset as the data source. However, even though this dataset is an excellent resource, it still represents just one platform; hence, research has to expand to others, to represent the current wide range of real-life MOOC environments, as different platforms have different structures and (acceptable) number of words per posts. To address this research gap and investigate other data sources, the present paper provides an analysis of the FutureLearn platform (which requires additional effort to complete the manual annotation). However, in common with the Stanford MOOC Post dataset, our dataset also suffers from similar disadvantages: identifying when instructor intervention is required from the massive number of posts in MOOC discussion forums is challenging for classifiers, due to the extremely limited number of urgent cases, which causes a highly imbalanced dataset. Moreover, as explained, correctly identifying the minority class (urgent comments) is the most important task. To date, and to the best of the researchers' knowledge, no research has targeted the problem of dealing with highly imbalanced data in the context of intervention in MOOCs.

2.2 Text augmentation

The other branch of prior research relevant to our paper is using text augmentation in NLP. The aim of text augmentation is to expand data (Liu et al. 2020), by providing and applying a set of techniques that create synthetic data from an existing dataset (Shorten et al. 2021). The performance of model predictions on a number of NLP tasks can be enhanced by text augmentation, and it is preventing overfitting (Li et al. 2022). It is used to alleviate the issue of limited or scarce labelled training data (Anaby-Tavor et al. 2020), which leads to low accuracy and recall for the minority class (Liu et al. 2020).

The existing literature shows that previous researchers utilised NLP augmentation approaches; for example, (Wang and Yang 2015) applied text augmentation by performing synonym replacement and identified similar words based on lexical and semantic embedding. Another study by (Kobayashi 2018) proposed a new word-based approach for text augmentation based on contextual augmentation; they applied synonym replacement, by using a bi-directional predictive language model. Next, (Wei and Zou 2019) explored straightforward text editing techniques for augmentation, using one of four simple techniques (synonym replacement, random insertion, random swap and random deletion). Recent work by Xiang et. al. 9Xiang et al. 2020) proposed a part-of-speech-focused lexical substitution for data augmentation (PLSDA) to generate more instances via word substitution. Another augmentation work is applied in translation: (Yu et al. 2018) generated new data to enhance their training data using back-translation with two translation models: the first translates sentences from English-to-French, while the second translates from French-to-English.

Some researchers tackled augmentation by using text augmentation libraries (NLPAug) for specific tasks. Jungiewicz and Smywiński-Pohl (2020) used a range of augmentation techniques for sentiment analysis, including (NLPAug) based on BERT and WordNet. More recently, (Pereira et al. 2021) used the same BERT-based library and contextual word-embedding augmenter to generate more programming problem statements on a training dataset.

In our current paper, we also augment the text data based on the (NLPAug) library. Unlike in prior research, usually focusing on word level for augmented data, we used in several different levels (character, word, sentence). We apply different techniques based on word embedding: word2vec (using words as a target), contextual word embedding: BERT, DistilBERT**,** RoBERTa, and XLNet (using words or sentences as a target), and OCR engine error (using characters as a target). In addition, we create various pipelines based on sequential flow. We construct three different approaches because in textual augmentation, the best approach is based on the dataset; if any approach improved on performance for specific data, this may have been detrimental to other data (Qiu et al. 2020).

2.3 Adaptive models in MOOCs

In this section, we present related works on adaptation and adaptive models implemented in MOOCs. As MOOCs are a rather recent addition, with the term ‘MOOC’ coined in 2008 (Stracke and Bozkurt 2008), and the ‘year of the MOOCs’ only launching them in 2012 (Jordan and Goshtasbpour 2022), adaptation has been slow to be introduced to them, with most of them still being designed via the ‘one-size-fits-all’ paradigm (Shimabukuro 2016; Rizvi et al. 2022), to some extent in spite of the decades of research in adaptive educational hypermedia (Ahmadaliev et al. 2019), intelligent tutoring system (Mousavinasab et al. 2021; Hodgson et al. 2021), and the like. Nevertheless, a few researchers have started proposing adaptation in MOOCs. For instance, (Alzetta et al. 2018) designed a customised learning path in an interactive and mobile learning environment and in MOOCs using a Question/Answering (QA) system. Another work on adaptive models in MOOCs (Lallé and Conati 2020) created a framework for user modelling and adaptation (FUMA), as an adaptive support to learners' during video usage. They used video watching and interaction behaviours as features, to reveal inactive learners. Another very recent work proposes an optimal learning path, to avoid MOOC learners from dropping out (SMAILI et al. 2022); they provide each learner an adaptive appropriate path, based on interaction with the environment, using particle swarm optimization (PSO).

In this research, unlike in previous research, we enable building *adaptive models based on learner comments*, with the aim is to improve communication with instructors.

3 Methodology

This study aims to automatically classify if a MOOC learner's comment is urgent and so requires flagging for instructor intervention. This means modelling learner data (their comments) to recommend an action to the instructor (here, reply). We call this a *fine-grained learner model*, as each learner is represented by the set of their comments. More formally, we can write that for learner l_1, their learner model L between time points t_1 and t_2 is given by:

$$L(l_1, t_1, t_2) = \{F_{t(c)\in[t1,t2]}(\mathrm{urgency}(l_1, c))\}$$

where F(.) can be any function aggregating the urgency for a given interval (e.g. a sum of urgency), and urgency (l_1,c) represents the fine-grained learner information at the level of a single comment c of learner l_1, made during the given time interval [t_1,t_2]. This learner model L(.) is used to drive the recommendation to instructors (see Sect. 3.3). To achieve this objective, we manually annotate a FutureLearn corpus; we additionally use the highly popular and well-used benchmark Stanford dataset to validate our best model, thus demonstrating generalisability for our approach, and applicability across courses and domains. More information on these datasets can be found in Sect. 3.1.

To determine the appropriate method, we use NLP techniques to construct a diverse predictive model for text classification. We employ two main types of supervised classifiers:

1. A traditional machine learning approach, with handcrafted features as a baseline model;
2. A fine-tuned version of BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al. 2018), representing the latest advance in NLP at the time of writing, as a powerful supervised deep learning model.

To tackle the imbalance problem, several different techniques were employed (see Sect. 3.2.2). One technique that we consider is text augmentation; here, we rely on different approaches (see Text Augmentation in Sect. 3.2.2) and augment the minority-class data with various multipliers (such as 3X and 9X). The reason for using text augmentation is that it prevents overfitting; it is considered a crucial regularisation technique (Coulombe 2018).

3.1 Datasets

This research was conducted on the FutureLearn and Stanford MOOC-based platform datasets.

3.1.1 Building UNITE: a Futurelearn-based dataset

FutureLearn, a European MOOC learning platform, is based on a *discussion in context* approach; comments are attached at each *course step* in the discussion area, excluding steps for quizzes and exercises (Chua et al. 2017). We collected comments written and posted by learners on a Big Data course, as a case study. This course was provided by Warwick University, United Kingdom. We selected this course due to its richness in comments, popularity, and the novelty of the subject, which would likely include an adequate number of urgent comments. Then, we prepared the data and manually annotated the dataset with the help of human experts, to create the Gold standard MOOC Urgency Corpus, a hand-labelled dataset. This task proved to be quite challenging even for the human experts. This confirms the findings of previous researchers: (Chandrasekaran et al. 2015a) noted that it is difficult for humans to create such a gold standard data set via manually labelling individual cases requiring instructor intervention.

Creating the gold standard dataset (UNITE) The corpus consists of 8263 comments (textual data in English) from the discussion forum of the above-mentioned course extracted over a 9-week period. Our research objectives are to classify urgent comments in discussion forums from the first half of the course, as our previous research indicated that most learners who dropped out were likely to do so in the early stages (Cristea et al. 2018; Alamri et al. 2019), so intervention, if required, would be likely be needed early on. In this regard, the following steps were taken to select suitable instances from the original data and prepare them for the annotation process. Learners' comments from the first half (weeks 1 to 5) of the course were extracted, representing approximately half of the 9-week course. After this point, all instructor comments were excluded. This resulted in a total of 5790 comments.

The annotation process was performed independently and manually by four computer science experts, three working as instructors in the Department of Computer Science at a different university to the authors (Kwara State University, Nigeria); additionally, the first author of the present paper was involved in labelling. In creating the Gold standard MOOC Urgency Corpus, we took a similar approach as that used for creating the Stanford dataset as on (Agrawal and Paepcke)'s website (https://datastage.stanford.edu/StanfordMoocPosts/) and in their research (Agrawal et al. 2015). Specifically, a Likert scale from 1–7 was used to classify the urgency of the comments: a value of 1 indicates that no reason exists for the instructor to read the post, while a value of 7 indicates extreme urgency (as shown in Fig. 1); for more information, see Sect. 3.1.2.

First, the data were pre-processed to exclude all comments with unmeaningful labels, such as (', 44, 0 and empty); this left a total of 5786 comments.

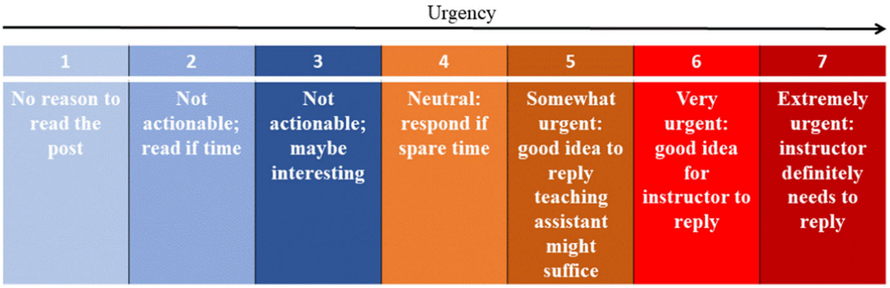
Urgency
1
2
3
4
5
6
7
No reason to read the post
Not actionable; read if time
Not actionable; maybe interesting
Neutral: respond if spare time
Somewhat urgent: good idea to reply teaching assistant might suffice
Very urgent: good idea for instructor to reply
Extremely urgent: instructor definitely needs to reply

Fig. 1 The scale of urgency applied (1–7)

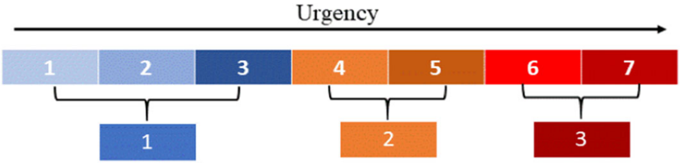

Fig. 2 Dimensionality reduction: converting the (1–7) scale into a (1–3) scale

Then, we validated and evaluated the quality of the manually labelled comments, by using the weighted Krippendorff's α (Antoine et al. 2014). The resulting agreement between annotators was low (α = 0.33); on the other hand, the Stanford dataset suffered partially from similar issues; the agreement between the optimal coder combination for the Likert variable (1–7) varies considerably per domain (Education: 0.14; Humanities/Sciences: 0.52; Medicine: 0.63).

Therefore, we first converted the (1–7) scale into a simplified (1–3) scale, as per Fig. 2. This meant, e.g. mapping 1, 2, 3 as non-urgent together—as they all are non-actionable, into (1). When recalculating the agreement, it remained, however, low (α = 0.31).

Thus, to be able to use the data reliably, we decided to identify a dependable sub-set; this sub-set was selected by including only comments that have a level of agreement between annotators of > 75%; in other words, at least 3 annotators (out of 4) must have agreed on the comment's label. Thus, we used a voting method, which is considered the most appropriate way to integrate different opinions about the same task (Troyano et al. 2004). In this case, only 4622 reliable comments could be included in the gold standard dataset (approximately 80% of the original data).

As we aimed to obtain as many potentially urgent comments as possible, we framed the problem as a binary classification problem, with outputs *Urgent* and *Non-urgent*, by converting and ranking the gold standard labels as:

- Scale = 2 or 3 → Urgent.
- Scale = 1 → Non-urgent.

Figure 3 depicts the final gold standard labels generated for this research. Please note that we erred on the side of caution in this final step by including *neutral* comments (urgency = 4) as *urgent*. This is because, for the Stanford data (Sect. 3.1.2), while

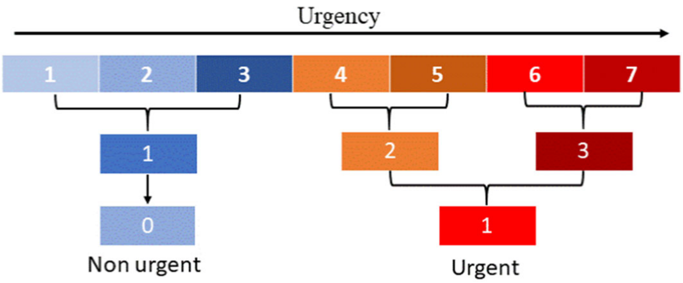

Fig. 3 Final gold standard labels for the UNITE corpus

some researchers supposed that *urgency* ≥ *4* represents *Urgent* comments (Almatrafi et al. 2018), others regard *urgency > 4* as *Urgent* (Guo et al. 2019). As here we were only working with integer values for labels, we considered that a value of *4 and above* signifies *Urgent*. This is also in line with our protocol on favouring recall and false positive (FP).

Therefore, *we define urgent comments as the comments that need response from instructors*. In general, the urgent comments can be about some specific problem encountered, or other latent causes, such as frustration, lack of knowledge, and change in circumstances.

Unsurprisingly, for our UNITE dataset, this division still resulted in a very high proportion of the comments being categorised as *non-urgent* (93%, i.e. 4,292 comments; with only 330 *urgent* comments 7%), showing a high degree of imbalance.

3.1.2 Stanford MOOC post dataset

The Stanford dataset (Agrawal et al. 2015) is a gold standard dataset available to academic researchers on request. It contains a large number of English learner forum posts (29,604 in total), commenting on 11 Stanford University MOOCs across three different domains (Humanities/Sciences, Medicine, and Education). Humanities/Sciences include six courses, Medicine four courses, and Education one course. The forum-post annotation was performed by three independent human coders: each post was manually labelled on six dimensions (*confusion, sentiment, urgency, question, answer* and *opinion). S*cores for *confusion, sentiment* and *urgency* were scored from 1 (low) to 7 (high). Meanwhile, the scores for the other items, *question, answer* and *opinion*, were classified using a binary scale (0 or 1). For more information, see (Agrawal and Paepcke) website (https://datastage.stanford.edu/StanfordMoocPosts/).

Similar to UNITE, for the Stanford dataset, the data were pre-processed by removing unmeaningful comments. This resulted in a total of 29,597 comments.

In the Stanford dataset, the urgency score (i.e. how urgent is it that an instructor reads the post) ranged from 1 = *non-urgent* to 7 = *very urgent* as shown in Fig. 1. However, in the current paper, we followed the classification detailed in Sect. 3.1.1: we framed the problem of detecting urgency as a binary classification task, by converting urgency into a binary value:

- Scale > 4 → Urgent.
- Remainder → Non-urgent.

We set our scale to > 4, because in the Stanford dataset, the label-calculating method does not produce an integer (1/1.5/2/2.5/3/3.5/4/4.5/5/5.5/6/6.5/7). This is further supported by our previous findings (Alrajhi et al. 2020), where we found a correlation between specific values (4 and 4.5) for the *sentiment* and *confusion* scales.

Ultimately, across the whole dataset, non-urgent cases represented 81% (23,991 comments) and urgent cases represented 19% (5606 comments—varying between 3.2 and 37.6% within their 11 courses) (with *urgent* posts having *urgency > 4*).

3.2 Experiments for imbalanced data

To achieve a comprehensive understanding of the best way to automatically identifying the urgency of comments on MOOCs, we use, as mentioned, two common supervised machine learning strategies (traditional shallow ML and deep learning—with BERT) to automatically classify the comments. Additionally, as urgency-detection is a typically imbalanced data problem; hence, any MOOC provider would need to take imbalance into account—we experiment with various techniques to deal with input data, as per Fig. 4.

First, we apply several training models to the original data on the gold standard UNITE corpus. Then, to improve performance, we design and develop three solutions to handle imbalanced data: (i) *text augmentation*; (ii) *text augmentation + undersampling*; and (iii) *undersampling* (see details in SubSect. 3.2.2). Text augmentation involves performing a range of approaches in different combinations, to augment the minority class in the training data. In undersampling, we randomly select instances from the majority class. Text augmentation + undersampling is a combinations of the two previous techniques. All the experiments were conducted using a stratified four fold cross-validation approach, to ensure representative results. The general architecture of the proposal classification model shown in Fig. 5 as we explained all the experiment in details in Sect. 3.2.

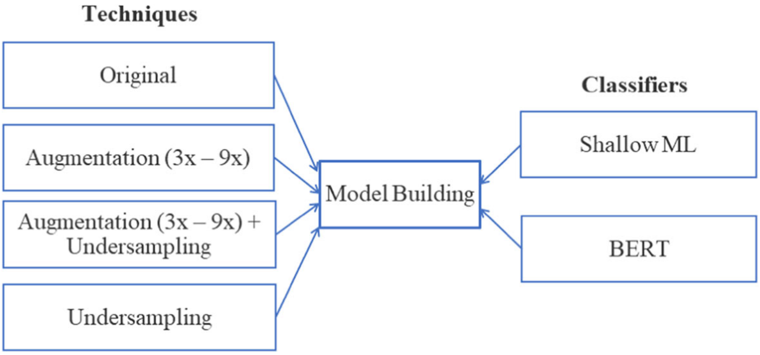

Fig. 4 Our proposed pre-processing (data balancing) and ML pipeline combinations

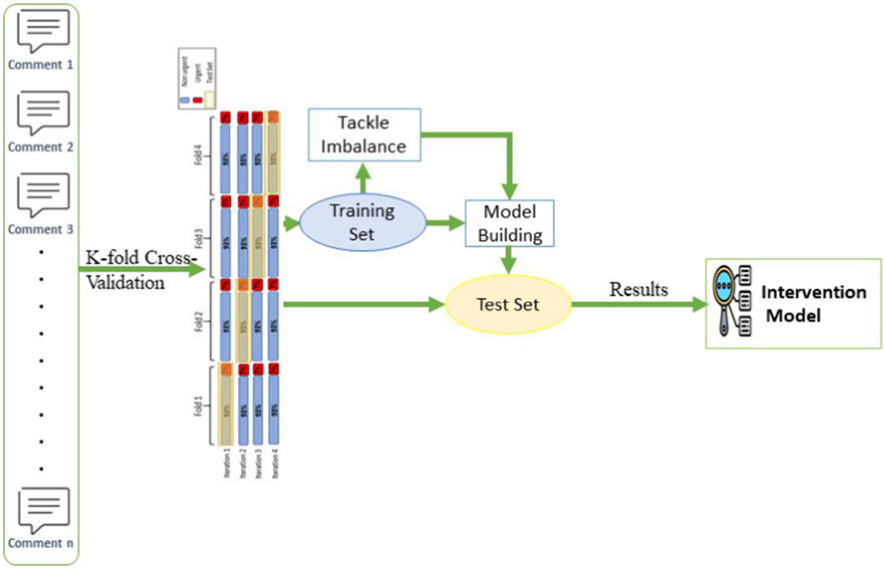

Fig. 5 The general architecture of the classification model

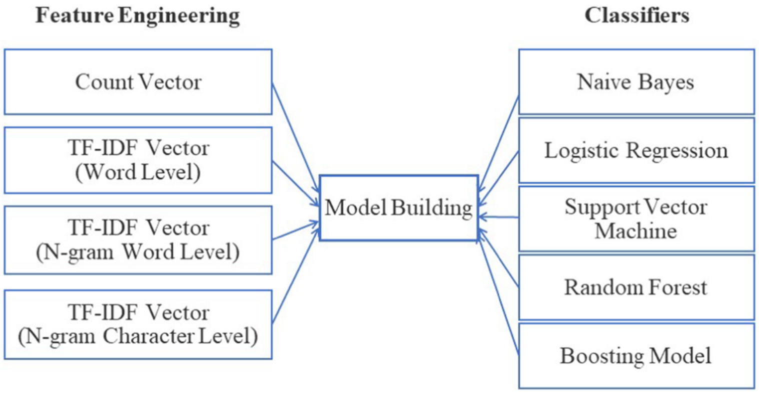

Fig. 6 Our framework of the shallow ML classifiers using different features

3.2.1 Classifiers

As said, we compare two major classification model types to classify the comments: (i) shallow machine learning (a basic model typically used by machine learning algorithms), and (ii) BERT (one of the most popular transformer models, as further explained).

Shallow machine learning We apply several machine learning models (see Fig. 6) to the classification task, each with different fundamental mechanisms for feature engineering, to capture the most effective features. This includes count vector and term frequency inverse document frequency (TF-IDF) to find an adequate classifier to predict urgent comments. We extract different feature sets via four different classical methods: (i) count vector; (ii) TF-IDF vector (word level); (iii) TF-IDF vector (n-gram word level); and (iv) TF-IDF vector (n-gram character level). Then, we build different popular classifiers across these different sets of features (naive Bayes, logistic regression, support vector machine, random forest, and boosting model—extreme (to become as gradient boosting (XGBoost)), as displayed in Fig. 6.

We represent each comment with a specific vector; the count-vector counts the frequency of every given word in every comment. TF-IDF calculates the score of a numerical statistic to evaluate the extent of relatedness between a particular word and a specific comment in a collection of comments; it thus represents a measure of how important a word is in a collection of comments. Three different levels of TF-IDF were considered as tokens (word, n-gram word with range (2,3) and n-gram character with a range of (2,3)) with *maximum features = 5000*.

BERT For deep learning, we employ the currently most popular and competitive approach in text classification tasks: BERT. Using BERT enabled us to avoid feature engineering, as well known for deep learning. We fine-tune a pre-trained 'BERT-Base, Uncased- (L = 12, H = 768, A = 12, Total Parameters = 110M)' version of the BERT classifier, which is the smaller model of the two available and was selected due to shorter training time, with one additional layer for the classification. For the BERT input, which is a sequence of tokens, we limit each comment to the final 128 tokens. This decision on the final tokens and size is based on various pre-experiment trials (final/first tokens; different sizes) that rendered this number (128 tokens) as the most suitable, encompassing most comments, with truncation only affecting 8% of UNITE, and 10% of the Stanford data. We use the Adam optimizer to tune BERT over 4 iterations.

3.2.2 Text balancing techniques

We developed several classifier models based on different techniques for manipulating the data. First, each of our models were run using the original gold standard corpus. Then, to tackle the imbalance problem, we independently applied the following approaches: (i) text augmentation; (ii) combined text augmentation then undersampling; and (iii) resampling using undersampling.

Original data usage (gold standard corpus) As an initial experiment, we implemented all our models directly with original UNITE data. We split the dataset into four groups using stratified k-fold cross-validation, choosing a value of $k = 4$ (4 folds). We chose the k-fold cross-validation run approach because it allowed us to obtain results with less bias to specific data (Berrar 2019). We use stratification in the dataset: the selection of data led to an equal distribution of every class in every set. Thus, every fold contained the same percentage of samples from each class (see Fig. 7) as follows: training fold 3466 or 3467 samples (3219 as class 0, i.e. non-urgent, and 247 or 248 as class 1); testing fold 1156 or 1155 samples (1073 as 0 and 83 or 82 as 1) in each iteration see Table 1. Please note that we did not use the more frequently

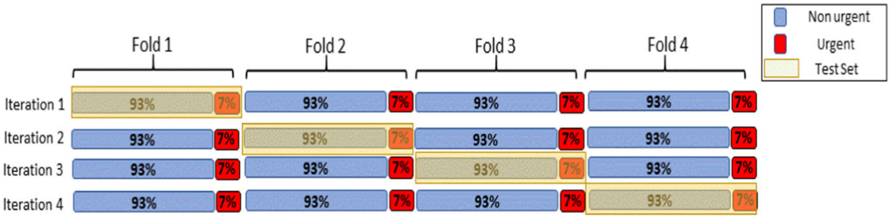

Fig. 7 Splitting the data using k-fold cross-validation and stratification

Table 1 Number of cases for every class in (training, test) sets in each iteration: original data

# of iteration	Training set		Test set	
	0	1	0	1
1	3219	247	1073	83
2	3219	247	1073	83
3	3219	248	1073	82
4	3219	248	1073	82

encountered ten fold validation, as, due to the very low number of urgent cases, this would have resulted in a too-low value per stratum for efficient stratification.

For the training with BERT, we divide the training data into 90% training (0 = 2897, 1 = 222 or 223) and 10% validation (0 = 322 1 = 25), as well as use stratification.

However, we found the results unsatisfactory for the various classifiers, see Sect. 4. This we considered was due to class imbalance. To overcome this issue and enhance prediction performance, we next employ alternative techniques, as in the next sections.

Text augmentation To manage the class imbalance and boost performance, we instead pre-process the data using artificial resampling (augmentation) to generate more minority-class cases for the training set of each fold, towards an almost balanced dataset. We augment every instance in the minority class into three and nine instances, respectively. We chose these values based on the literature reporting that for some databases, a low number of repetitions might not be sufficient to decrease the bias of the model in indiscriminately predicting the majority class; however, a higher repetition value might also render the data non-representative (Haixiang et al. 2017; Madabushi et al. 2020; Fonseca et al. 2020), so experimentation is necessary. Thus, in our work, we experiment, at every iteration, with the number of items in the training and test set for 3X and 9X augmentation, as in Table 2.

To achieve the augmentation goal, we apply common, easy-to-implement techniques for text augmentation, using the public (NLPAug) library. The text augmentation library (NLPAug) is a Python library dedicated to augmentation (Raghu and Schmidt 2020). We accessed simple code via the Edward Makcedward Github repository (Makcedward 2020). We use 3 different hybrid approaches: (i) word-level with

Table 2 Number of cases for every class in (training, test) sets in each iteration: text augmentation (3X–9X)

Quantities	# of iteration	Training set		Test set	
		0	1	0	1
3X	1	3219	988	1073	83
	2	3219	988	1073	83
	3	3219	992	1073	82
	4	3219	992	1073	82
9X	1	3219	2470	1073	83
	2	3219	2470	1073	83
	3	3219	2480	1073	82
	4	3219	2480	1073	82

Table 3 The approaches using different augmenters

Approach	Level	Augmenter	Type	Action
1	Word	ContextualWordEmbsAug	BERT	Insert
				Substitute
			DistilBERT	Substitute
2	Word	WordEmbsAug	Word2vec	Substitute
		ContextualWordEmbsAug	BERT	Substitute
			RoBERTa	Substitute
3	Character	OcrAug	OCR	Substitute
	Word	ContextualWordEmbsAug	BERT	Substitute
	Sentence	ContextualWordEmbsForSentenceAug	XLNet	Insert

the same type (BERT), (ii) word-level with different types; and (iii) different levels (character, word, sentence), as shown in Table 3.

In the first, we apply a hybrid approach that consists of three different actions (3X) in a ContextualWordEmbsAug augmenter based on BERT, by inserting and substituting with BERT and substituting with DistilBERT, to discover the most appropriate word for augmentation, as shown in Table 4.

Then, we build upon the (3X) method and increase the number of instances to (9X), by generating an additional 3X more instances for every instance. This is achieved by constructing six sequential pipelines, each representing a multi-augmenter (bi- or tri-augmenter), as shown in Table 5. Table 6 provides examples of 9X augmentation.

Next, we conduct the second approach, another augmentation procedure, by mixing several augmenter functions based on word-level (see Table 3): WordEmbsAug (substitute word2vec) and ContextualWordEmbsAug (substitute BERT and substitute RoBERTa).

Table 4 An example of different augmenters for 3X in the first approach on a comment in UNITE

Type	Text
Original	I hope any course staff member can help us to solve this confusion asap!!!
BERT (insert)	i hope any course support staff member can come help enable us to solve this current confusion case asap ! ! !
BERT (substitute)	our trust one important staff member can help us to solve this confusion slowly ! ! !
DistilBERT (substitute)	i hope any course faculty member should teach us to alleviate problem confusion asap ! ! !

Table 5 Different pipelines to generate (9X) in the first approach

Pipeline	Type	Action
Pipeline 1	BERT	Insert
	BERT	Substitute
Pipeline 2	BERT	Insert
	DistilBERT	Substitute
Pipeline 3	BERT	Substitute
	BERT	Insert
	DistilBERT	Substitute
Pipeline 4	BERT	Substitute
	DistilBERT	Substitute
Pipeline 5	DistilBERT	Substitute
	BERT	Substitute
	BERT	Insert
Pipeline 6	DistilBERT	Substitute
	BERT	Insert

Last, as per Table 3, we construct the third approach, which is based on three different levels of augmenter (character, word and sentence). For character-level, we used OcrAug (a substitute for OCR). For word-level, we used ContextualWordEmbsAug (a substitute for BERT). For sentence-level, we used ContextualWordEmbsForSentenceAug (insert XLNet).

Then, we apply the shallow machine learning and BERT models, as explained in Sect. 3.2.1, based on 3X and 9X augmentations.

Text augmentation + undersampling By creating nine new artificial instances in the training set, we obtain an almost-balanced dataset, albeit with a concern about its non-representativity. However, by creating three new instances, we moderately increase the data variation and perform a smaller move towards balancing the dataset. Hence, we address the concern of minimising model errors by frequently predicting

Table 6 An example of different augmenters for 9X in the first approach

Type	Text
Original	I hope any course staff member can help us to solve this confusion asap!!!
BERT (insert)	i hope any acting course staff member can help us financially to solve these this … confusion situation asap ! ! !
BERT (substitute)	i recommend a course staff member can help our all solve this confusion tonight ! ! !
DistilBERT (substitute)	i hope any helpful staff member may help us to unlock the mystery asap ! ! !
Pipeline 1	the four know some successful course change group member can even get us this solve this global confusion asap ! ! !
Pipeline 2	as i hope any course staff experienced can somehow help inspire us and suggest solving this puzzle asap ! ! !
Pipeline 3	i wonder if various further course instructors or volunteers could employ you might ultimately solve this particular trouble indeed ! ! !
Pipeline 4	we hope only one staff volunteer to help us both solve their confusion immediately ! ! !
Pipeline 5	sincerely hope any new permanent staff department member cannot aid me by easily in solve this time at crisis ! ! !
Pipeline 6	and i hope for any Facebook staff member can persuade them to quickly solve this situation well together ! ! !

the majority class, achieving instead high accuracy yet low recall and precision for the minority class. We deal with these two concerns by applying a hybrid resampling method combining this augmentation technique with undersampling.

In these experiments, we aim to balance the datasets by combining both text augmentation and undersampling methods as follows. First, by increasing instances to 3X or 9X in the minority class. Second, in undersampling, we randomly reduce the number of elements in the majority class to be equal to the minority class in every fold. Therefore, the numbers of samples for each pipeline in the *urgent* and *non-urgent* classes were approximately 990 for 3X and 2475 for 9X.

Undersampling (random) To balance the class distribution in the original data, we performed an alternative popular method—the undersampling technique for imbalanced data classification—by randomly removing instances in the majority class. Thus, in this case, the numbers of samples for each class were 247 or 248.

Future learn and Stanford datasets As explained, the distribution of the *urgent* class in the FutureLearn dataset (7%) was different than in the Stanford dataset (19%). Therefore, the effect of these different techniques to handle imbalanced data was expected to affect the performance results of the different datasets. Figures 8 and 9 show the distribution of every class in every fold for every method for UNITE and (3X) for the Stanford dataset, respectively.

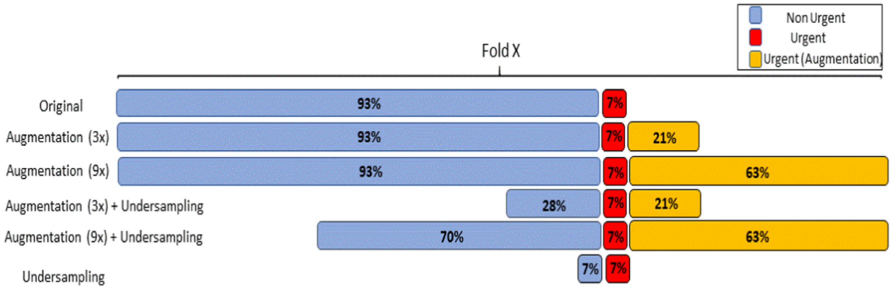

Fig. 8 The distribution of every class in every fold in every method for UNITE: our FutureLearn dataset

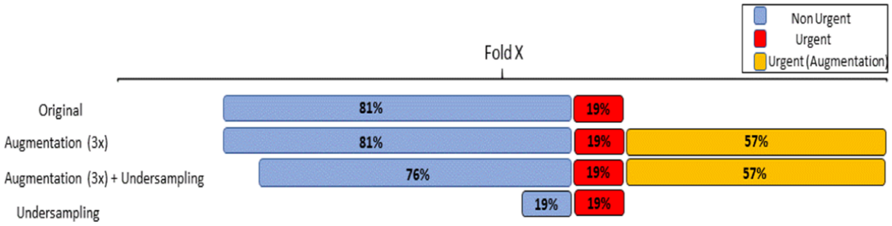

Fig. 9 The distribution of every class in every fold for every method for the Stanford dataset

3.3 Illustration of adaptive intervention models

In this section, we introduce the design of illustrative adaptive intervention models for instructor interaction, based on our automatic urgency detection. These models showcase how the user model parameters proposed by this study can fit in simpler or, gradually, more complex user models; users here mean *instructors*, as primary target users, and *learners*, as potential secondary target users. Specifically, we provide two practical scenarios for semi-automatic instructor intervention: (1) semi-automatic intervention that tackles unbalanced data with a classification model. (2) filtering comments that improve instructor intervention, by filtering the results based on learners, their number of comments and time of posting the comment.

3.3.1 Semi-automatic instructor intervention: basic scenario

The first scenario introduces an artificial supporting instructor, as a pipeline incorporating the classification model, representing the learner model, using additional information on the instructor (the instructor model), as shown in Fig. 10.

A basic instructor model would minimally contain variables such as the instructor time available for a specific session, and a time of reading per comment, or, alternatively, a maximum number of comments to read in that session (hence, a simple 2-variable user model for the instructor). The learner model contains 2-variables as well: comments of learners and urgency of comments at post-level (fine-grained). Based on this information, the adaptive intervention model can automatically retrieve

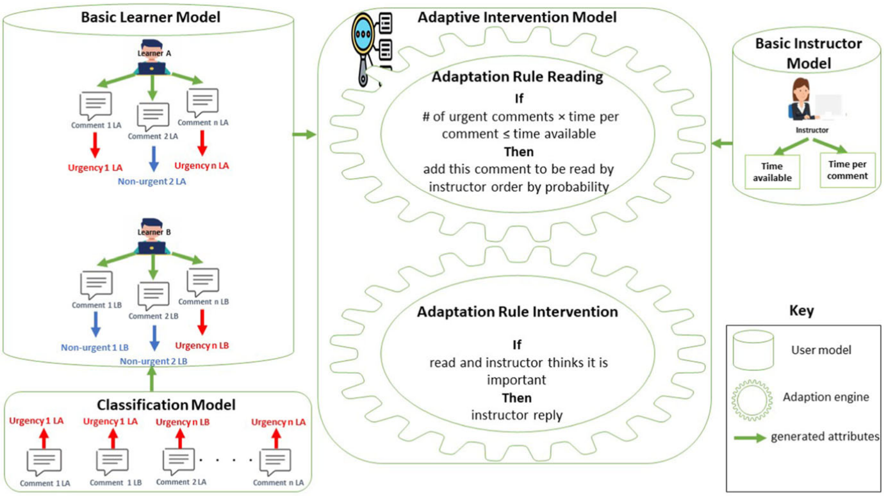

Fig. 10 The adaptive intervention model based on learners' comments; note how our predicted urgency becomes a (derived, fine-grained) learner model variable, together with the comments per learner

the topmost urgent comments, depending on their ranking (e.g. from a probability score given by a classification model) and thus, reducing the overload on the instructor.

For example, instructor Laura has answered all yesterday's comments from learners. She wishes to know if there are any urgent comments today, as she has only 30 min, after which she needs to go to teach another course. All this information represents the instructor model. The MOOC webpage for today has 3 items, and each has acquired a total of, in average, 150 comments. She thinks that she would be able to answer about 10–15 comments maximum (and adds this information to her instructor model[1]). Thus, the artificial support instructor recommends Laura to answer the most urgent top 5 comments for each of the 3 items from today's class. This recommendation represents the adaptive model, which is the combination of the classification model and our technique to deal with imbalanced data, automatically classifying posts and detecting urgent comments, adapting to the instructor's needs, and helping Laura avoid reading all the comments, and improving her interaction with the learners.

3.3.2 Semi-adaptive instructor intervention: expanded scenario based on coarse granularity and expanded learner models

The first scenario deals with the recommended urgent comments, as per our pipeline proposed in this paper. However, this model can be further improved. Next, we show how comments can be grouped, to further refine the learner model, and deal with urgency at (higher granularity) *learner level*, instead of the *comment level*. This may show if a learner is generally in trouble and needs support, which may make dealing with that learner more stringent. That is consistent with findings in the study of (Alrajhi

[1] Alternatively, the system could automatically convert Laura's available time (of 30 min) into a number of questions to be answered.

et al. 2021), which showed that learners write more comments overall when they require urgent intervention.

For example, instructor John wishes to use Laura's system for classifying comments, but has noticed that learners tend to either send many urgent comments, when they are in trouble, or are overall happy, and thus, send fewer comments. He would like his load reduced and hence not answer to seemingly urgent comments coming from users with very few comments. Thus, he wishes learners to be grouped into urgent and non-urgent learner, as shown in Fig. 11. John will now be able to answer first to the *urgent learners*, even if some of the *non-urgent learners* may have posted comments sounding as urgent, but who may be less needing intervention.

An extension to the learner model would be to add this coarse-grained, learner-level classification to the learner model, the number of comments, and then to further cluster them based on it. We compute the correlation between the number of written comments per learner versus the number of comments from these that need urgent intervention, using Pearson's correlation. Therefore, we apply first Silhouette analysis, to check the number of clusters, then Fisher–Jenks algorithm (because we only work on one-dimensional data), perform the clustering. These clusters are then merged into 2 groups that differentiate between high number of comments learners and low number ones (urgent/non-urgent learners).

In addition, we further adapt the intervention based on the time stamp of the comments of each of these learners, to provide John with comments of the urgent learners, ordered first-come-first-served (FCFS). Thus, number of comments and time stamps are variables added to the extended learner model in this example. The overall adaptive model is summarised in Fig. 12, using the same instructor model as previously, but an expanded, three-variable learner model: coarse-grained learner-level urgency; with fine-grained post-level urgency; and learner comments.

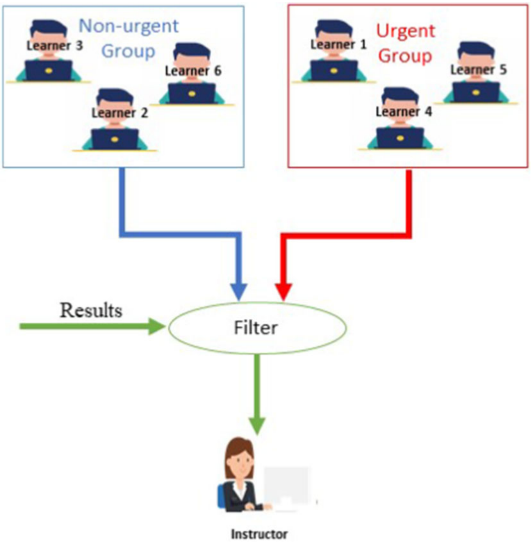

Fig. 11 Refining the learning modelling of urgency, based on two groups (non-urgent/urgent)

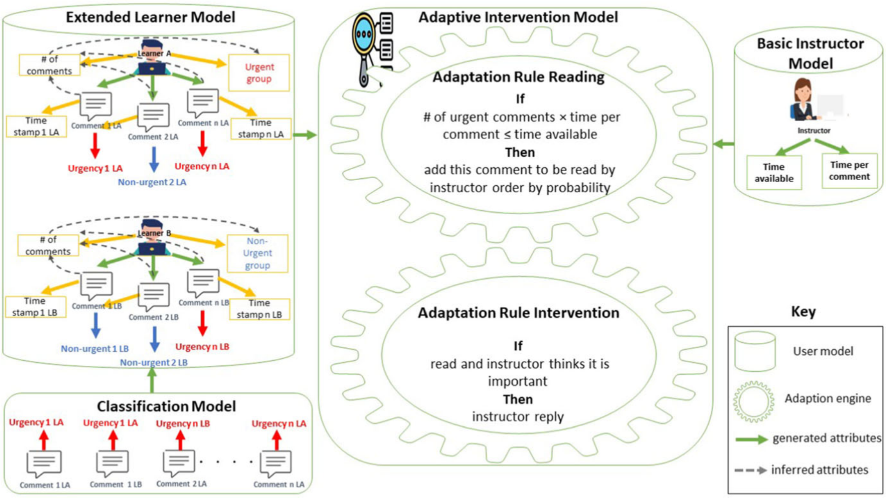

Fig. 12 The adaptive intervention model based on coarse-grained, expanded learner modelling, with two learner groups based on number of comments (low/high); here, the instructor model is the same as in Fig. 10, but the learner model has been expanded with an additional variable

4 Results

This section provides a discussion of our experimental results for the two main types of classifiers (shallow machine learning and deep learning) as well results related to our example adaptation intervention models.

4.1 Experiments for imbalanced data

4.1.1 Shallow machine learning on the UNITE dataset

In shallow machine learning, five different classifiers were tested with different types of feature engineering in different augmentation approaches (Approach #1: word-level, with the same type (BERT), Approach #2: word-level with different types (Word2vec, BERT and RoBERTa) and Approach #3: different levels (character, word, sentence) with different types (OCR, BERT and XLNet) as discussed in Sect. 3.2.2 under text augmentation heading. Table 7 shows the results of the comparison between the accuracy (ACC) of the basic classifier (naive Bayes) with count vector as features. Despite some of these models obtaining around 90% accuracy (see Table 7), this does not mean that they are good models; they could be biased towards the majority class on the imbalanced class dataset. Thus, to achieve more accurate results, we used other metrics to measure performance, such as Precision (*P*), Recall (*R*) and F1 Measure (*F1*) derived from the true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negatives (*FN*) of the confusion matrix.

Table 7 Results of the Naive Bayes model with count-vector feature engineering with original data, with 3 approaches to augmentation (see Table 3) using 3X and 9X (Table 2) with and without undersampling and with undersampling without augmentation. Underline: best R for class 1 (Urgent), Bold: best performance, balancing between class 1 (Urgent) and class 0 (Non-urgent) in the UNITE dataset

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
Count vector	×		×	0.92	0.93	0.99	0.96	0.29	0.05	0.08
	Approach #1	3X	×	0.90	0.94	0.95	0.94	0.26	0.24	0.25
		9X	×	0.84	0.95	0.88	0.91	0.21	0.44	0.29
		3X	√	0.75	0.96	**0.76**	0.85	0.16	**0.57**	0.25
		9X	√	0.81	0.96	0.84	0.89	0.19	0.50	0.28
	Approach #2	3X	×	0.91	0.94	0.97	0.95	0.29	0.18	0.22
		9X	×	0.90	0.94	0.95	0.95	0.25	0.21	0.23
		3X	√	0.79	0.96	**0.81**	0.88	0.17	**0.51**	0.26
		9X	√	0.88	0.94	0.93	0.94	0.24	0.28	0.26
	Approach #3	3X	×	0.90	0.94	0.96	0.95	0.28	0.21	0.24
		9X	×	0.87	0.95	0.92	0.93	0.23	0.31	0.26
		3X	√	0.78	0.96	**0.80**	0.87	0.17	**0.55**	0.26
		9X	√	0.85	0.95	0.89	0.92	0.20	0.36	0.25
	×		√	0.52	0.97	0.49	0.65	0.11	0.82	0.19

This research project aims to correctly classify *urgent* cases, which is represented by Recall *R*. We propose to use *Recall* as the main evaluation metric for *urgent* comments, as Recall (the correct identification of most of the urgent cases, preferably all, allowing for false positives) ensures that all urgent cases have precedence, which is more important than Precision (the correct identification of only *urgent* cases, but possibly missing some, allowing for false negatives). Specifically, we try to improve the outcome of *R* for the positive class. In addition, we separately show how we can add a filtering process, to retrieve the most urgent comments that obtained priority from their probability on classification models. Thus, we potentially reduce the instructor effort required to review and read many comments.

Table 7 shows the count-vector feature as a case study. The evaluation of *R* for class 1 (*urgent*), based on the original data, was very low (0.05). We were able to improve performance, by applying different approaches to enhance the data and address the imbalance problem. The best result was obtained using undersampling (Under), which was a significant improvement (0.82; this improvement is statistically significant: Mann–Whitney U test: $p < 0.05$), but the results dramatically decreased for the class 0 to 0.49 (from 0.99). In contrast, the performance of the manipulated data with 3X augmentation + undersampling achieved the best performance, balancing between class 1 and class 0. Most of the three approaches for augmentation have the same scenario. There are some exceptions, which will be discussed later in this section.

Our aim is to find the best techniques to deal with the imbalanced data between three different approaches for augmentation (not to find the best feature engineering approach). The reason to use different features is to confirm which imbalanced data technique is better across all feature sets and to make our experiments more general. Therefore, *we can generalise the findings* to (a) *all approaches on specific features*, (b) *all features on a specific classifier*, and (c) *all classifiers*, since the effectiveness of the proposed methods of data manipulation was similar for most classifiers (as shown in Appendix B). We decided to report and discuss only one of these classifiers (naive Bayes) with one feature (count vector), for conciseness; the results of the other types of classifiers are provided in Appendix B. However, the exceptions are discussed in the next paragraph.

Whilst most of the findings are the same, there are a few exception cases; for example, (1) The strongest predictors for recall (R) were mostly those with undersampling (Under). However, some models (Random Forest and Boosting) with TF-IDF vectors (n-gram word level) are better in other approaches for text augmentation than undersampling (Table 8). (2) The best performance was often obtained from the data with 3X augmentation + undersampling, achieving a balance between class 1 and class 0 levels, but there are some models 9X augmentation + undersampling that outperform 3X augmentation + undersampling, as shown in Table 9. (3) in terms of approaches, the goal of building more than one approach was to generalise the results of the technique used in data manipulation. Thus, the results for different approaches reveal that there is no approach we can consider as best. However, interestingly, approach 3, which is based on different levels (character, word, sentence), always has the best results for R if we use TF-IDF vectors (n-gram character level) as a feature, across all experiments (as shown in Appendix B).

4.1.2 BERT on the UNITE dataset

When using BERT, Table 10 shows the prediction performance for the different methods of manipulating the data. As mentioned, only augmentation was performed; no feature engineering was necessary. The performance of R for class 1 in BERT with the original data was not too low in comparison with the shallow machine learning results. Although rose from (0.52) to $\underline{0.82}$ with the undersampling technique, this difference is statistically significant (Mann–Whitney U test: $p < 0.05$). However, for the negative class, recall decreased from 0.98 to 0.86. To achieve more balance between the two classes, we used 3X augmentation + undersampling (see Table 10).

Hence, the best classifier performance on the UNITE dataset is *BERT with the 'approach 3 with* 3X *augmentation + undersampling'*.

To verify the effectiveness of the different data manipulation techniques to deal with the imbalanced data problem, we utilised the same methods on the Stanford dataset. In these experiments, we limited augmentation to 3X only, since 9X would have generated more instances in the minority class than in the majority class. We also applied only 'approach 3' (see Table 3), which provided the best performance for the (3X augmentation + undersampling) technique on the UNITE dataset.

Table 8 Cases in which the results of the text augmentation techniques are higher than the results of undersampling technique

Classifier	Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
						P	R	F1	P	R	F1
Random forest	TF-IDF vectors (n gram word level)	Approach #1	9X	×	0.91	0.95	0.95	0.95	0.36	0.39	0.38
			3X	√	0.90	0.95	0.94	0.95	0.33	0.40	0.36
			9X	√	0.89	0.95	0.93	0.94	0.30	0.41	0.34
		Approach #2	3X	√	0.89	0.95	0.93	0.94	0.30	0.39	0.34
			9X	√	0.88	0.95	0.93	0.94	0.27	0.36	0.31
		Approach #3	9X	×	0.91	0.95	0.95	0.95	0.38	0.41	0.39
			3X	√	0.90	0.95	0.94	0.95	0.34	0.41	0.37
			9X	√	0.90	0.95	0.93	0.94	0.33	0.42	0.37
		×		√	0.86	0.95	0.90	0.92	0.21	0.36	0.26
Boosting		Approach #1	9X	√	0.70	0.95	0.71	0.81	0.12	0.53	0.20
		Approach #2	9X	√	0.66	0.95	0.67	0.79	0.11	0.55	0.19
		×		√	0.74	0.95	0.76	0.84	0.14	0.49	0.21

Table 9 Cases in which the results of the 9X augmentation + undersampling are higher than the results of 3X augmentation + undersampling

Classifier	Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
						P	R	F1	P	R	F1
SVM	TF-IDF vectors (n gram word level)	Approach #1	3X	√	0.91	0.95	0.95	0.95	0.34	0.33	0.34
			9X	√	0.89	0.95	0.93	0.94	0.27	0.35	0.31
		Approach #2	3X	√	0.90	0.95	0.95	0.95	0.31	0.30	0.30
			9X	√	0.88	0.95	0.93	0.94	0.25	0.32	0.28
		Approach #3	3X	√	0.92	0.94	0.97	0.96	0.38	0.20	0.26
			9X	√	0.91	0.94	0.97	0.95	0.35	0.24	0.28
Random forest		Approach #1	3X	√	0.90	0.95	0.94	0.95	0.33	0.40	0.36
			9X	√	0.89	0.95	0.93	0.94	0.30	0.41	0.34
		Approach #3	3X	√	0.90	0.95	0.94	0.95	0.34	0.41	0.37
			9X	√	0.90	0.95	0.93	0.94	0.33	0.42	0.37
Boosting		Approach #1	3X	√	0.85	0.95	0.88	0.92	0.20	0.38	0.26
			9X	√	0.70	0.95	0.71	0.81	0.12	0.53	0.20
		Approach #2	3X	√	0.77	0.95	0.80	0.87	0.14	0.42	0.21
			9X	√	0.66	0.95	0.67	0.79	0.11	0.55	0.19
		Approach #3	3X	√	0.85	0.95	0.89	0.92	0.20	0.38	0.27
			9X	√	0.87	0.95	0.91	0.93	0.25	0.40	0.31

Table 10 Results of the BERT model with original data, with 3 approaches to augmentation (see Table 3) using 3X and 9X (Table 2) with and without undersampling and with undersampling without augmentation. Underline: best R for class 1 (Urgent), Bold: best performance, balancing between class 1 (Urgent) and class 0 (Non-urgent) in the UNITE dataset

Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
				P	R	F1	P	R	F1
×		×	0.95	0.96	0.98	0.97	0.67	0.52	0.58
Approach #1	3X	×	0.95	0.97	0.97	0.97	0.62	0.63	0.63
	9X	×	0.94	0.97	0.96	0.97	0.54	0.59	0.57
	3X	√	0.92	0.98	**0.93**	0.96	0.46	**0.75**	0.57
	9X	√	0.93	0.97	0.95	0.96	0.50	0.63	0.56
Approach #2	3X	×	0.95	0.97	0.97	0.97	0.62	0.62	0.62
	9X	×	0.94	0.97	0.97	0.97	0.57	0.57	0.57
	3X	√	0.91	0.98	**0.92**	0.95	0.41	**0.77**	0.54
	9X	√	0.94	0.97	0.96	0.97	0.55	0.62	0.58
Approach #3	3X	×	0.94	0.97	0.97	0.97	0.60	0.59	0.59
	9X	×	0.94	0.97	0.97	0.97	0.61	0.59	0.60
	3X	√	0.89	0.98	**0.89**	0.94	0.36	**0.79**	0.50
	9X	√	0.94	0.97	0.97	0.97	0.57	0.58	0.58
×		√	0.86	0.98	0.86	0.92	0.32	0.82	0.46

4.1.3 BERT on the Stanford dataset

Table 11 shows the results of BERT on the Stanford dataset. We obtained similar results for the UNITE data; the only difference being in the performance of the two techniques with 3X augmentation with and without undersampling. This is possibly because the distribution of *non-urgent* cases differs between the two datasets (see Figs. 8, 9). Whereas, as we clarified in Fig. 9, the distribution of *non-urgent* cases for 3X is almost the same as the distribution of *non-urgent* cases for (3X + undersampling).

Table 11 Results of the BERT model with original data, with 3 approaches to augmentation (see Table 3) using 3X (Table 2) with and without undersampling and with undersampling without augmentation. Underline: best R for class 1 (Urgent), Bold: best performance, balancing between class 1 (Urgent) and class 0 (Non-urgent) in the Stanford dataset

Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
				P	R	F1	P	R	F1
×		×	0.91	0.94	0.96	0.95	0.80	0.73	0.76
Approach #3	3X	×	0.91	0.95	**0.94**	0.94	0.75	**0.78**	0.77
	3X	√	0.91	0.95	**0.94**	0.95	0.76	**0.78**	0.77
×		√	0.89	0.97	0.89	0.93	0.65	0.89	0.75

Table 12 Results of the Naive Bayes model with count vector as a feature engineering with first approaches to augmentation (see Table 3) using 3X with undersampling. First row: basic model with all data, second row: filtering model with top urgent 5 comments for the urgent class '1' in the UNITE dataset

# Comments	1		
	P	R	F1
All	0.16	0.56	0.25
5	0.40	1.00	0.57

4.2 Adaptive intervention models

4.2.1 Basic adaptation scenario

In this scenario, depending on urgent comments ranking (probability score given by the classification model), the aim is for the adaptive intervention model to automatically retrieve the most important urgent comments and reduce the number of comments that are read by instructor. In this case, we used naïve Bayes with count vector, using approach #1 for 3X augmentation with undersampling model (the best performance among different approaches in naïve Bayes with count vector) as a case study. For example, if the time available is limited to read 5 comments, then the model will retrieve only 5 comments. Table 12 presents the results of the comparison between the *basic model* (all comments) and the *adaptive model* that selects just the top urgent (5) comments for the urgent class (1), which clearly outperforms the basic model on all evaluation criteria.

4.2.2 Expanded adaptation scenario

For the second scenario, we proposed an adaptation filtering model based on the number of learner comments. We use Pearson's correlation to calculate the correlation between the number of written comments per learner and the number of comments from those that require immediate attention. This process resulted in a strong correlation (0.65).

The results of Fisher–Jenks algorithm to cluster learners are shown in Table 13. To obtain the two groups (for urgent/ non-urgent learners), we then merge clusters 1 and 2, to reflect the learners with a high number of comments, as these are significantly more communicative than learners in cluster 0.

Table 13 Clustering learners based on their number of comments

Cluster	Count	Mean	SD	Min	Max
0	734	3.30	3.06	1	15
1	57	27.26	12.50	16	62
2	6	107.16	34.31	84	173

Table 14 Number of comments on Test Set; First row: basic models, second row: filtering models in the UNITE dataset

Fold	Model	Number of comments on test set
1	Basic	1156
	Filtering	533
2	Basic	1156
	Filtering	561
3	Basic	1155
	Filtering	551
4	Basic	1155
	Filtering	552

We remove comments from the low number of comments group (non-urgent learners) from each fold (using stratified fourfold cross-validation). The number of comments on the test set is shown in Table 14 for both: *basic* model, which contains all learners; and the *filtering* model, which only contains learners with a high number of comments (urgent learners). Hence, the number of comments in the filtering model is much lower than for the basic model. For example, in fold 1 it dropped from (1156 to 533) basic to filtering, reducing the number of comments the instructor needs to read. Thus, whilst the overall recall is somewhat reduced (by 11%), the load of the instructor is significantly ($p < < 0.5$) reduced as well.

5 Error analysis

We conducted an in-depth re-analysis of our model to understand the reasons for the errors obtained in the test set in every fold. For this purpose, we manually inspected the examples of mistakes that our best algorithm (BERT—Text Augmentation + Undersampling) made on UNITE data. Specifically, *false negatives* (*FN*), which the model categorised as *non-urgent* (although they are labelled as urgent), were considered to be the most critical errors, as our aim was to capture *all* urgent cases. To put the results and especially the errors in context, we compared the miss-predictions of the classifier with human-level performance for the different folds (using stratified k-fold cross-validation, choosing a value of $k = 4$ (4 folds) as we explained in methodology under SubSect. 3.2.2. The results are shown in Table 15.

Table 15 FN results for the best algorithm versus disagreement between human annotators

Fold	FN (total)	Human disagreement
1	23	19
2	14	11
3	19	16
4	14	13

Table 16 Anonymised examples of FN results and disagreement between human annotators on UNITE data

Fold	Example
1	I have some difficulties to understand diagrams. But it seems very important to give a meaning and a context to words used in analysis
2	I had done this, the [programming-platform] is going on. But I need also the [other-platform]. I installed a old Version [other-platform], also the Newest. I couldn't found the [other-platform] for [setup]
3	Further to my comment on the previous "step" I am yet to be convinced!
4	I don't understand the reason of this message when I type [code-removed] Warning message:[error-message-removed]

From the results, we found that most of the FN cases were also mirrored in the disagreement between annotators (i.e. for 19/23 false negatives misclassified by our classifier, the human annotators also disagreed for fold 1, etc., see Table 15). This further supports the notion that decision-making among annotators is difficult, as well as that the more difficult cases are both hard for humans and classifiers to categorise, examples of each fold are shown in Table 16.

From Table 16, we can better understand why humans and ML struggle in certain cases. For example, in fold 1, the learner does not understand the diagram, but s/he is happy about providing a meaning and context for the words used in analysis. Some annotators believe that this comment is not urgent, because the learner did not request assistance. However, another annotator may find that the learner has difficulty in understanding the concept. Such clashes may explain why the model was not able to detect the above-mentioned urgent cases.

6 Discussion, limitations and future work

In MOOC environments, detecting the *urgent cases* is a critical issue. As per the nature of MOOCs, urgent cases are rare, compared to non-urgent ones, which leads to unbalanced data. Also, the other issue in MOOCs is that the intervention in past researches (Almatrafi et al. 2018; Guo et al. 2019) follows an one-size-fits-all approach, without any personalisation in the intervention based on learners, in spite of long-term personalisation research in education.

In this paper, our aim is to propose a *solution for unbalanced data based on MOOCs and adapt and improve the system interaction by automating urgency detection that would enable an instructor to decide when to react*, so adapting the timing of interventions to the urgency detected in the learner. The potential beneficiaries are MOOC providers, then instructors on MOOCs, then the learners.

The ultimate goal of this work is the last step, where we personalise (by automatic adaptation) the identification process of urgent comments in MOOCs for the instructors, our primary users. This means we are building 'interactive computer systems that can be adapted or adapt themselves to their current users', adapting to the needs of our primary users, the instructors, to manage their workload, as well as, indirectly, to the needs of our secondary users, the learners, to have their urgent messages identified (and ultimately, answered).

Urgency in intervention is an interesting area, raising the question: how dependent is urgency on the learner? for instance, for the latter, it is possible that another learner has already dealt with the urgent question. So, showing the full thread to the instructor to inspect is also useful. Here, we look first at *fine-grained learner modelling*, where we consider each comment as a feature of a learner, that needs, if urgent, to be dealt with on its own. Next, as research has revealed correlations between urgency and number of comments, showing that learners posting urgent comments are likely to post many of them, hence being able to be classified, at the macro-scale, as an 'urgent learner' (Alrajhi et al. 2021), we also propose *coarse-grained learner modelling*, where learners are grouped as either *urgent-learners* or *non-urgent learners*. Such learners would need to be treated with priority by the instructors.

Modelling learners based on comments only is a simplification of the learner model, as in any model is a simplification of the world. However, we believe that the comments of learners can provide insight into some of the learner characteristics and needs. For instance, the language of the comment can show anxiety, or a certain level of background knowledge, or impatience, thus covering various learner model variables. It is, however, possible some learners are missed this way; moreover, learners that refuse to engage with comments will not be identified via these methods.

Learner models can contain several parameters, and be simpler or richer. Indeed, learner models can reflect various aspects of a learner, and they often include various parameters, such as current level of confusion, motivation, and understanding. Interventions to reduce drop out of the learner from the MOOC could include changing the difficulty or type of problems, referring the learner to modules for missing prerequisite knowledge, peer referrals, encouraging communications, etc. In this paper, we add to this rich tapestry of user model dimensions, by extracting urgency based directly on user comments, which have not been considered before in user modelling. Importantly, we consider comment-based user modelling rich, in the sense that they may reflect various aspects of a learner—boredom, interest, knowledge, fluency, etc. Our learner model can be used by itself, or in conjunction with other parameters of the users (if known), and thus, further enrich the user model. This, however, does not detract from the merit of the parameters we introduce with our approach.

The main limitation is that the automatic classification in general and the potential solution for unbalanced data may not be general enough for all online courses and platforms, as it has been applied on only one specific course in FutureLearn. However, as showed in Sect. 3, we have further validated our best solution on the highly popular and well-used Stanford dataset, thus strengthening the case for generalisability for our approach, and applicability across courses and domains.

There are numerous opportunities for future work such as: exploiting other features, like the number of posts in a thread; while this may not directly tell us if an individual post is urgent, we could analyse numbers per topic, or per learner, etc. Such multitude of posts may only reflect, however, a very popular topic, or a very prolific learner. Interestingly, analysing the correlation between FutureLearn 'likes' of posts and their urgency showed no correlation between them. Moreover, as not all posts have 'likes', our current approach is more generalizable. Also, an urgency lexicon-based method, based on the identification of key terms (keywords or n-grams) that could indicate urgency may be considered; however, current deep learning methods are known to outperform lexicon-based ones. Another interesting approach would be to increase priority in urgent case based on number of learners; i.e. if an issue is raised by many learners, it could be considered of higher urgency. In addition, peer reactions could indeed be taken into account both in terms of declaring a problem solved (i.e. a peer has answered it) or generating a flurry of responses thus being very urgent (as many in the class would struggle with the same issue).

Additionally, detection of learner affective states may allow (artificial) instructors to adapt their support to those states. Furthermore, labelling data based on sentiment analysis, and especially confusion/frustration perspective, as a supervised, but, more interestingly, as an unsupervised method, may be a cheaper way to detect urgency—however, may raise challenges in terms of accuracy. Our data were already labelled for urgency, regardless of the original cause—frustration, lack of knowledge, change in circumstances, etc. Thus, we believe our approach to be more generic, as it encompasses all these reasons or other latent causes. Please finally also note that urgency is a relative concept; we have addressed some of these aspects in this paper, within its definition (Sect. 3.1.1); however, further work can look into refining or specialising its definition.

Finally, whilst our results are very specific to MOOC comment analysis, our techniques may serve as a template for other similar NLP classification tasks using machine learning with severely skewed datasets.

7 Conclusion

On MOOC platforms, deciding the right moment for instructor intervention is an important challenge to be overcome to better support learners and lower drop-out rates. Building an automated model to detect comments that require urgent intervention represents a promising solution to this problem. However, the available comment datasets naturally contain only a few urgent cases, leading to imbalanced data, which explains the difficulty in creating models to detect such cases accurately. In this work, we analysed and compared three techniques (text augmentation, text augmentation + undersampling, and undersampling) to improve the quality of such data. Also, we provided several new pipelines incorporating different text augmenters. Our results show that an increase in model performance can be obtained via undersampling, and a combination of text augmentation + undersampling achieves the best performance in balancing between the two classes.

These results help in retrieving the most urgent comments for instructors. To show how this can be applied, we have illustrated it with two adaptive models, based on two types of user models: (1) personalised instructor intervention based on a fine-granularity learner model and (2) filtering results based on a higher granularity learner model.

We further inspected wrongly classified urgent instances and found that the problem does not simply lie with the classifier: it also stems from the data, which humans also find difficult to annotate. This indicates that the difficulties faced by human annotators in classifying such commentsare also faced by these models.

Additionally, whilst the majority of previous works on instructor intervention were based on the Stanford corpus, in this research, we used the FutureLearn platform featuring a total of 5790 comments annotated by human experts, to form the new UNITE corpus.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [Laila] and [Ahmed]. Programming by [Laila], [Filipe] and [Ahmed]. The first draft of the manuscript was written by [Laila], [Ahmed], [Filipe] and [Alexandra]. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Declarations

Competing interests The authors declare no competing interests.

Appendix A

The metrics that we used in this research to measure performance, Precision (P), Recall (R) and F1 Measure (*F1*) derived from the true positive (TP), true negative (TN), false positive (FP), and false negatives (FN) of the confusion matrix, calculated as:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$F1 = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{TN} + \mathrm{FP}}$$

Appendix B

The results on naive Bayes with other feature engineering and the other shallow models (logistic regression, support vector machine, random forest and boosting model—extreme gradient boosting (XGBoost)) rendered similar results as those shown in the results section (naive Bayes model with count vector as a feature engineering) in the UNITE dataset (Tables 17, 18, 19, 20 and 21).

Table 17 Results of the Naive Bayes model with various types of feature engineering with original data, with 3 approaches to augmentation (see Table 3) using 3X and 9X (Table 2) with and without undersampling and with undersampling without augmentation in the UNITE dataset

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
TF-IDF vectors (word level)	×		×	0.93	0.93	1.00	0.96	0.00	0.00	0.00
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.62	0.05	0.10
		9X	×	0.89	0.94	0.94	0.94	0.26	0.26	0.26
		3X	√	0.77	0.96	0.79	0.86	0.17	0.57	0.26
		9X	√	0.84	0.95	0.87	0.91	0.20	0.43	0.28
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.69	0.05	0.10
		9X	×	0.91	0.94	0.97	0.95	0.31	0.20	0.25
		3X	√	0.79	0.96	0.81	0.88	0.18	0.53	0.27
		9X	√	0.88	0.95	0.92	0.93	0.24	0.32	0.27
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.72	0.05	0.10
		9X	×	0.91	0.94	0.96	0.95	0.30	0.21	0.24
		3X	√	0.80	0.96	0.82	0.89	0.19	0.52	0.27
		9X	√	0.87	0.95	0.92	0.93	0.24	0.35	0.28
	×		√	0.47	0.98	0.44	0.60	0.11	0.90	0.19
TF-IDF vectors (n gram word level)	×		×	0.93	0.93	1.00	0.96	1.00	0.00	0.01
	Approach #1	3X	×	0.93	0.94	0.99	0.96	0.52	0.13	0.21
		9X	×	0.90	0.95	0.94	0.95	0.30	0.31	0.31

Table 17 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		3X	√	0.86	0.95	0.89	0.92	0.24	0.45	0.31
		9X	√	0.87	0.95	0.91	0.93	0.24	0.39	0.30
	Approach #2	3X	×	0.93	0.94	0.99	0.96	0.52	0.12	0.20
		9X	×	0.90	0.95	0.95	0.95	0.31	0.29	0.30
		3X	√	0.85	0.95	0.89	0.92	0.23	0.45	0.30
		9X	√	0.88	0.95	0.92	0.93	0.26	0.36	0.30
	Approach #3	3X	×	0.93	0.94	0.99	0.96	0.47	0.13	0.20
		9X	×	0.90	0.95	0.95	0.95	0.30	0.30	0.30
		3X	√	0.86	0.95	0.89	0.92	0.24	0.42	0.30
		9X	√	0.87	0.95	0.91	0.93	0.25	0.38	0.30
	×		√	0.63	0.97	0.62	0.76	0.13	0.72	0.22
TF-IDF vectors (n gram character level)	×		×	0.93	0.93	1.00	0.96	0.31	0.02	0.03
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.57	0.06	0.11
		9X	×	0.92	0.94	0.99	0.96	0.39	0.12	0.19
		3X	√	0.88	0.95	0.92	0.94	0.28	0.40	0.33
		9X	√	0.91	0.94	0.97	0.95	0.34	0.20	0.25

Table 17 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.57	0.06	0.11
		9X	×	0.93	0.93	0.99	0.96	0.41	0.08	0.13
		3X	√	0.90	0.95	0.95	0.95	0.32	0.33	0.33
		9X	√	0.92	0.93	0.99	0.96	0.35	0.10	0.16
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.56	0.07	0.12
		9X	×	0.93	0.94	0.99	0.96	0.48	0.14	0.21
		3X	√	0.89	0.96	0.92	0.94	0.31	0.44	0.36
		9X	√	0.92	0.94	0.98	0.96	0.39	0.18	0.25
	×		√	0.56	0.98	0.53	0.69	0.13	0.87	0.22

Table 18 Results of the Logistic Regression model with various types of feature engineering with original data, with 3 approaches to augmentation (see Table 3) using 3X and 9X (Table 2) with and without undersampling and with undersampling without augmentation in the UNITE dataset

Feature Engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
Count vector	×		×	0.92	0.94	0.98	0.96	0.38	0.14	0.21
	Approach #1	3X	×	0.91	0.94	0.96	0.95	0.34	0.24	0.28
		9X	×	0.89	0.95	0.94	0.94	0.28	0.32	0.30
		3X	√	0.84	0.95	0.87	0.91	0.21	0.45	0.29
		9X	√	0.88	0.95	0.92	0.93	0.26	0.38	0.31
	Approach #2	3X	×	0.91	0.94	0.96	0.95	0.33	0.25	0.28
		9X	×	0.89	0.95	0.94	0.94	0.28	0.30	0.29
		3X	√	0.83	0.96	0.86	0.90	0.21	0.51	0.30
		9X	√	0.88	0.95	0.92	0.93	0.26	0.36	0.30
	Approach #3	3X	×	0.91	0.94	0.97	0.95	0.35	0.24	0.28
		9X	×	0.90	0.94	0.95	0.95	0.28	0.28	0.28
		3X	√	0.84	0.96	0.87	0.91	0.23	0.50	0.32
		9X	√	0.88	0.95	0.93	0.94	0.26	0.34	0.29
	×		√	0.71	0.96	0.72	0.82	0.15	0.64	0.24
TF-IDF vectors (word level)	×		×	0.93	0.93	1.00	0.96	0.00	0.00	0.00
	Approach #1	3X	×	0.93	0.94	0.99	0.96	0.49	0.15	0.24

Table 18 (continued)

Feature Engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	×	0.89	0.95	0.93	0.94	0.29	0.35	0.32
		3X	√	0.83	0.96	0.86	0.91	0.22	0.52	0.31
		9X	√	0.87	0.95	0.90	0.93	0.25	0.44	0.32
	Approach #2	3X	×	0.93	0.94	0.99	0.96	0.56	0.16	0.25
		9X	×	0.91	0.95	0.96	0.95	0.34	0.29	0.31
		3X	√	0.83	0.96	0.85	0.90	0.22	0.55	0.31
		9X	√	0.88	0.95	0.92	0.94	0.26	0.35	0.30
	Approach #3	3X	×	0.93	0.94	0.99	0.96	0.52	0.16	0.24
		9X	×	0.90	0.95	0.95	0.95	0.31	0.30	0.30
		3X	√	0.85	0.96	0.88	0.91	0.23	0.49	0.31
		9X	√	0.88	0.95	0.92	0.93	0.27	0.40	0.32
	×		√	0.72	0.97	0.72	0.83	0.16	0.68	0.26
TF-IDF vectors (n gram word level)	×		×	0.93	0.93	1.00	0.96	1.00	0.00	0.01
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.69	0.09	0.17
		9X	×	0.90	0.95	0.94	0.94	0.30	0.35	0.32
		3X	√	0.86	0.96	0.89	0.92	0.25	0.46	0.32

Table 18 (continued)

Feature Engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	√	0.86	0.95	0.89	0.92	0.24	0.45	0.32
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.60	0.08	0.14
		9X	×	0.90	0.95	0.94	0.94	0.29	0.32	0.30
		3X	√	0.85	0.96	0.88	0.92	0.23	0.46	0.31
		9X	√	0.85	0.95	0.88	0.92	0.22	0.43	0.29
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.65	0.09	0.16
		9X	×	0.90	0.95	0.95	0.95	0.33	0.34	0.33
		3X	√	0.87	0.95	0.91	0.93	0.26	0.43	0.33
		9X	√	0.88	0.95	0.91	0.93	0.28	0.42	0.33
	×		√	0.74	0.96	0.75	0.84	0.16	0.62	0.25
TF-IDF vectors (n gram character level)	×		×	0.93	0.93	1.00	0.96	1.00	0.01	0.01
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.65	0.04	0.07
		9X	×	0.93	0.93	0.99	0.96	0.58	0.09	0.16
		3X	√	0.93	0.95	0.97	0.96	0.46	0.28	0.35
		9X	√	0.93	0.94	0.99	0.96	0.53	0.12	0.19
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.72	0.05	0.10
		9X	×	0.93	0.93	1.00	0.96	0.64	0.09	0.16

Table 18 (continued)

Feature Engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
	Approach #3	3X	√	0.93	0.95	0.97	0.96	0.48	0.32	0.39
		9X	√	0.93	0.94	0.99	0.96	0.58	0.14	0.22
		3X	×	0.93	0.94	1.00	0.97	0.71	0.13	0.22
		9X	×	0.93	0.94	0.99	0.97	0.63	0.16	0.26
		3X	√	0.91	0.95	0.94	0.95	0.36	0.41	0.38
		9X	√	0.93	0.94	0.99	0.96	0.57	0.25	0.34
	×		√	0.75	0.97	0.76	0.85	0.18	0.69	0.29

Table 19 Results of the Support Vector Machine model with various types of feature engineering with original data, with 3 approaches to augmentation (see Table 3) using 3X and 9X (Table 2) with and without undersampling and with undersampling without augmentation in the UNITE dataset

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
Count vector	×		×	0.93	0.93	1.00	0.96	0.00	0.00	0.00
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.62	0.08	0.14
		9X	×	0.91	0.94	0.96	0.95	0.34	0.25	0.28
		3X	√	0.88	0.95	0.92	0.93	0.25	0.36	0.30
		9X	√	0.89	0.95	0.94	0.94	0.27	0.29	0.28
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.57	0.07	0.12
		9X	×	0.92	0.94	0.98	0.96	0.39	0.20	0.26
		3X	√	0.86	0.95	0.90	0.92	0.22	0.37	0.27
		9X	√	0.91	0.94	0.96	0.95	0.33	0.27	0.29
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.57	0.08	0.13
		9X	×	0.92	0.94	0.98	0.96	0.42	0.21	0.28
		3X	√	0.87	0.95	0.91	0.93	0.25	0.38	0.30
		9X	√	0.91	0.95	0.96	0.95	0.35	0.29	0.32
	×		√	0.66	0.95	0.66	0.78	0.12	0.58	0.19
TF-IDF vectors (word level)	×		×	0.93	0.93	1.00	0.96	0.00	0.00	0.00
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.64	0.09	0.15

Table 19 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	×	0.92	0.94	0.98	0.96	0.38	0.20	0.26
		3X	√	0.91	0.95	0.96	0.95	0.34	0.29	0.32
		9X	√	0.91	0.94	0.96	0.95	0.33	0.25	0.28
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.59	0.09	0.15
		9X	×	0.93	0.94	0.98	0.96	0.46	0.20	0.28
		3X	√	0.90	0.95	0.94	0.95	0.32	0.34	0.33
		9X	√	0.92	0.94	0.97	0.96	0.38	0.24	0.29
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.71	0.07	0.13
		9X	×	0.93	0.94	0.99	0.96	0.51	0.16	0.25
		3X	√	0.92	0.94	0.97	0.96	0.38	0.26	0.31
		9X	√	0.93	0.94	0.98	0.96	0.46	0.21	0.29
	×		√	0.72	0.96	0.73	0.83	0.16	0.65	0.25
TF-IDF vectors (n gram word level)	×		×	0.93	0.93	1.00	0.96	0.50	0.00	0.01
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.56	0.07	0.12

Table 19 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	×	0.91	0.95	0.96	0.95	0.34	0.28	0.30
		3X	√	0.91	0.95	0.95	0.95	0.34	0.33	0.34
		9X	√	0.89	0.95	0.93	0.94	0.27	0.35	0.31
	Approach #2	3X	×	0.93	0.93	0.99	0.96	0.55	0.08	0.14
		9X	×	0.91	0.94	0.96	0.95	0.31	0.23	0.26
		3X	√	0.90	0.95	0.95	0.95	0.31	0.30	0.30
		9X	√	0.88	0.95	0.93	0.94	0.25	0.32	0.28
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.55	0.05	0.09
		9X	×	0.92	0.94	0.98	0.96	0.41	0.18	0.25
		3X	√	0.92	0.94	0.97	0.96	0.38	0.20	0.26
		9X	√	0.91	0.94	0.97	0.95	0.35	0.24	0.28
	×		√	0.67	0.96	0.66	0.79	0.14	0.68	0.23
TF-IDF vectors (n gram character level)	×		×	0.93	0.93	1.00	0.96	1.00	0.00	0.01
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.67	0.02	0.05
		9X	×	0.93	0.93	1.00	0.96	0.74	0.06	0.11
		3X	√	0.93	0.94	0.99	0.97	0.61	0.23	0.33
		9X	√	0.93	0.93	1.00	0.96	0.73	0.08	0.15

Table 19 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.73	0.05	0.09
		9X	×	0.93	0.93	1.00	0.96	0.68	0.07	0.13
		3X	√	0.94	0.95	0.99	0.97	0.60	0.27	0.37
		9X	√	0.93	0.94	1.00	0.96	0.67	0.11	0.19
	Approach #3	3X	×	0.94	0.94	1.00	0.97	0.81	0.13	0.22
		9X	×	0.93	0.94	1.00	0.97	0.70	0.14	0.24
		3X	√	0.93	0.95	0.97	0.96	0.51	0.34	0.41
		9X	√	0.94	0.94	0.99	0.97	0.69	0.18	0.29
	×		√	0.76	0.97	0.76	0.85	0.18	0.70	0.29

Table 20 Results of the Random Forest model with various types of feature engineering with original data, with 3 approaches to augmentation (see Table 3) using 3X and 9X (Table 2) with and without undersampling and with undersampling without augmentation in the UNITE dataset

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
Count vector	×		×	0.93	0.93	1.00	0.96	0.80	0.01	0.02
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.54	0.04	0.07
		9X	×	0.93	0.94	0.98	0.96	0.46	0.19	0.27
		3X	√	0.88	0.95	0.91	0.93	0.27	0.41	0.33
		9X	√	0.90	0.95	0.95	0.95	0.33	0.34	0.34
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.64	0.05	0.09
		9X	×	0.92	0.94	0.98	0.96	0.40	0.14	0.21
		3X	√	0.87	0.95	0.90	0.93	0.24	0.41	0.31
		9X	√	0.92	0.94	0.97	0.96	0.37	0.25	0.30
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.67	0.04	0.08
		9X	×	0.93	0.93	0.99	0.96	0.42	0.09	0.15
		3X	√	0.90	0.95	0.93	0.94	0.32	0.40	0.35
		9X	√	0.92	0.94	0.97	0.96	0.41	0.25	0.31
	×		√	0.68	0.96	0.68	0.80	0.14	0.68	0.23
TF-IDF vectors (word level)	×		×	0.93	0.93	1.00	0.96	0.67	0.01	0.01
	Approach #1	3X	×	0.93	0.93	1.00	0.96	0.62	0.04	0.07

Table 20 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	×	0.92	0.94	0.98	0.96	0.41	0.20	0.27
		3X	√	0.88	0.95	0.92	0.94	0.29	0.43	0.35
		9X	√	0.89	0.95	0.93	0.94	0.28	0.34	0.31
	Approach #2	3X	×	0.93	0.93	1.00	0.96	0.58	0.05	0.08
		9X	×	0.92	0.94	0.98	0.96	0.41	0.20	0.27
		3X	√	0.87	0.95	0.91	0.93	0.25	0.41	0.31
		9X	√	0.90	0.95	0.95	0.95	0.32	0.33	0.32
	Approach #3	3X	×	0.93	0.93	1.00	0.96	0.67	0.04	0.08
		9X	×	0.93	0.94	0.99	0.96	0.50	0.18	0.27
		3X	√	0.89	0.95	0.93	0.94	0.31	0.41	0.36
		9X	√	0.92	0.95	0.97	0.96	0.42	0.32	0.36
	×		√	0.71	0.96	0.71	0.82	0.14	0.63	0.24
TF-IDF vectors (n gram word level)	×		×	0.93	0.93	1.00	0.96	0.59	0.05	0.09
	Approach #1	3X	×	0.92	0.94	0.98	0.96	0.45	0.23	0.31

Table 20 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	×	0.91	0.95	0.95	0.95	0.36	0.39	0.38
		3X	√	0.90	0.95	0.94	0.95	0.33	0.40	0.36
		9X	√	0.89	0.95	0.93	0.94	0.30	0.41	0.34
	Approach #2	3X	×	0.93	0.95	0.98	0.96	0.47	0.26	0.34
		9X	×	0.91	0.95	0.95	0.95	0.33	0.31	0.32
		3X	√	0.89	0.95	0.93	0.94	0.30	0.39	0.34
		9X	√	0.88	0.95	0.93	0.94	0.27	0.36	0.31
	Approach #3	3X	×	0.93	0.94	0.98	0.96	0.48	0.25	0.32
		9X	×	0.91	0.95	0.95	0.95	0.38	0.41	0.39
		3X	√	0.90	0.95	0.94	0.95	0.34	0.41	0.37
		9X	√	0.90	0.95	0.93	0.94	0.33	0.42	0.37
	×		√	0.86	0.95	0.90	0.92	0.21	0.36	0.26
TF-IDF vectors (n gram character level)	×		×	0.93	0.93	1.00	0.96	0.69	0.03	0.05
	Approach #1	3X	×	0.93	0.93	0.99	0.96	0.48	0.06	0.11
		9X	×	0.93	0.93	0.99	0.96	0.34	0.05	0.08
		3X	√	0.93	0.94	0.98	0.96	0.51	0.22	0.31
		9X	√	0.93	0.93	0.99	0.96	0.43	0.07	0.12
	Approach #2	3X	×	0.93	0.93	0.99	0.96	0.38	0.06	0.10
		9X	×	0.93	0.93	0.99	0.96	0.38	0.08	0.13

Table 20 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
	Approach #3	3X	√	0.93	0.95	0.98	0.96	0.46	0.26	0.34
		9X	√	0.92	0.93	0.99	0.96	0.40	0.10	0.16
		3X	×	0.93	0.94	0.99	0.96	0.57	0.14	0.23
		9X	×	0.93	0.94	0.99	0.96	0.51	0.15	0.23
		3X	√	0.92	0.95	0.97	0.96	0.46	0.33	0.39
		9X	√	0.93	0.94	0.99	0.96	0.48	0.18	0.26
	×		√	0.67	0.97	0.66	0.79	0.15	0.75	0.24

Table 21 Results of the Boosting model with various types of feature engineering with original data, with 3 approaches to augmentation (see Table 3) using 3X and 9X (Table 2) with and without undersampling and with undersampling without augmentation in the UNITE dataset

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
Count vector	×		×	0.93	0.93	1.00	0.96	0.74	0.04	0.08
	Approach #1	3X	×	0.92	0.94	0.98	0.96	0.43	0.17	0.24
		9X	×	0.88	0.95	0.92	0.93	0.24	0.34	0.28
		3X	√	0.81	0.96	0.83	0.89	0.19	0.51	0.27
		9X	√	0.83	0.95	0.86	0.91	0.20	0.43	0.27
	Approach #2	3X	×	0.93	0.94	0.99	0.96	0.48	0.17	0.25
		9X	×	0.91	0.94	0.96	0.95	0.34	0.25	0.29
		3X	√	0.81	0.96	0.83	0.89	0.19	0.53	0.28
		9X	√	0.89	0.95	0.93	0.94	0.27	0.32	0.29
	Approach #3	3X	×	0.92	0.94	0.98	0.96	0.41	0.14	0.21
		9X	×	0.91	0.94	0.96	0.95	0.31	0.25	0.28
		3X	√	0.82	0.95	0.84	0.89	0.19	0.48	0.27
		9X	√	0.88	0.95	0.92	0.93	0.26	0.36	0.30
	×		√	0.72	0.96	0.73	0.83	0.14	0.58	0.23

Table 21 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
TF-IDF vectors (word level)	×		×	0.93	0.93	1.00	0.96	0.67	0.04	0.08
	Approach #1	3X	×	0.93	0.94	0.98	0.96	0.45	0.17	0.25
		9X	×	0.87	0.95	0.91	0.93	0.23	0.33	0.27
		3X	√	0.80	0.96	0.83	0.89	0.18	0.51	0.27
		9X	√	0.83	0.95	0.86	0.91	0.20	0.45	0.28
	Approach #2	3X	×	0.93	0.94	0.99	0.96	0.45	0.15	0.23
		9X	×	0.91	0.95	0.96	0.95	0.37	0.27	0.31
		3X	√	0.79	0.95	0.82	0.88	0.16	0.47	0.24
		9X	√	0.89	0.95	0.93	0.94	0.28	0.34	0.31
	Approach #3	3X	×	0.93	0.94	0.99	0.96	0.50	0.16	0.25
		9X	×	0.91	0.94	0.96	0.95	0.32	0.26	0.29
		3X	√	0.81	0.95	0.84	0.89	0.18	0.48	0.27
		9X	√	0.88	0.95	0.91	0.93	0.25	0.38	0.30
	×		√	0.69	0.96	0.70	0.81	0.13	0.60	0.22
TF-IDF vectors (n gram word level)	×		×	0.93	0.93	1.00	0.96	0.62	0.04	0.07
	Approach#1	3X	×	0.93	0.94	0.99	0.96	0.51	0.14	0.22

Table 21 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	×	0.92	0.95	0.96	0.96	0.40	0.31	0.35
		3X	√	0.85	0.95	0.88	0.92	0.20	0.38	0.26
		9X	√	0.70	0.95	0.71	0.81	0.12	0.53	0.20
	Approach #2	3X	×	0.93	0.94	0.99	0.96	0.54	0.15	0.23
		9X	×	0.92	0.94	0.97	0.96	0.40	0.26	0.32
		3X	√	0.77	0.95	0.80	0.87	0.14	0.42	0.21
		9X	√	0.66	0.95	0.67	0.79	0.11	0.55	0.19
	Approach #3	3X	×	0.93	0.94	0.99	0.96	0.51	0.15	0.23
		9X	×	0.90	0.95	0.95	0.95	0.32	0.30	0.31
		3X	√	0.85	0.95	0.89	0.92	0.20	0.38	0.27
		9X	√	0.87	0.95	0.91	0.93	0.25	0.40	0.31
	×		√	0.74	0.95	0.76	0.84	0.14	0.49	0.21
TF-IDF vectors (n gram character level)	×		×	0.93	0.94	0.99	0.97	0.65	0.15	0.24
	Approach #1	3X	×	0.93	0.94	0.99	0.97	0.61	0.20	0.31
		9X	×	0.93	0.94	0.99	0.96	0.53	0.19	0.28
		3X	√	0.92	0.95	0.96	0.96	0.44	0.39	0.42

Table 21 (continued)

Feature engineering	Augmentation		Under	Acc	Non-urgent 0			Urgent 1		
					P	R	F1	P	R	F1
		9X	√	0.93	0.94	0.98	0.96	0.53	0.24	0.33
	Approach #2	3X	×	0.93	0.94	0.99	0.97	0.61	0.21	0.31
		9X	×	0.93	0.94	0.99	0.96	0.53	0.18	0.26
		3X	√	0.91	0.95	0.95	0.95	0.39	0.40	0.39
		9X	√	0.93	0.94	0.98	0.96	0.47	0.23	0.31
	Approach #3	3X	×	0.93	0.95	0.98	0.96	0.56	0.29	0.39
		9X	×	0.93	0.95	0.98	0.96	0.49	0.29	0.37
		3X	√	0.91	0.96	0.94	0.95	0.38	0.51	0.44
		9X	√	0.92	0.95	0.97	0.96	0.46	0.36	0.40
	×		√	0.77	0.97	0.78	0.86	0.18	0.65	0.29

References

Agrawal, A., Paepcke, A.: The stanford moocposts data set. https://Datastage.Stanford.Edu/Stanfordmoocposts/

Agrawal, A., Venkatraman, J., Leonard, S., Paepcke, A.: Youedu: addressing confusion in MOOC discussion forums by recommending instructional video clips. In: The 8th international conference on educational data mining (2015).

Ahmadaliev, D.K., Medatov, A.A., Jo'rayev, M.M., O'rinov, N.T.: Adaptive educational hypermedia systems: an overview of current trend of adaptive content representation and sequencing. Theoret. Appl. Sci. **3**, 58–61 (2019)

Alamri, A., Alshehri, M., Cristea, A., Pereira, F. D., Oliveira, E., Shi, L., Stewart, C. Predicting MOOCS dropout using only two easily obtainable features from the first week's activities. In: International Conference on Intelligent Tutoring Systems, 2019. Springer, 163–173.

Almatrafi, O., Johri, A.: Systematic review of discussion forums in massive open online courses (Moocs). IEEE Trans. Learn. Technol. **12**, 413–428 (2018)

Almatrafi, O., Johri, A., Rangwala, H.: Needle in a haystack: identifying learner posts that require urgent response in mooc discussion forums. Comput. Educ. **118**, 1–9 (2018)

Alrajhi, L., Alharbi, K., Cristea, A. I.: A multidimensional deep learner model of urgent instructor intervention need in mooc forum posts. In: International Conference On Intelligent Tutoring Systems, Springer, 226–236 (2020)

Alrajhi, L., Alamri, A., Pereira, F. D., Cristea, A. I.: Urgency analysis of learners' comments: An automated intervention priority model for mooc. In: International Conference On Intelligent Tutoring Systems, Springer, 148–160 (2021)

Alzetta, C., Adorni, G., Celik, I., Koceva, F., Torre, I.: Toward a user-adapted question/answering educational approach. In: Adjunct Publication Of The 26th Conference On User Modeling, Adaptation and Personalization, 173–177 (2018)

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., Zwerdling, N.: Do not have enough data? Deep learning to the rescue! In: Proceedings Of The Aaai Conference On Artificial Intelligence, 7383–7390 (2020)

Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J.: Engaging with massive online courses. In: Proceedings of the 23rd international conference on world wide web, 687–698 (2014)

Antoine, J.-Y., Villaneau, J., Lefeuvre, A.: Weighted Krippendorff's alpha is a more reliable metrics for multi-coders ordinal annotations: experimental studies on emotion, opinion and coreference annotation. In: Eacl 2014, 10p (2014)

Bakharia, A.: Towards cross-domain mooc forum post classification. In: Proceedings of the Third (2016) ACM Conference On Learning@ Scale, ACM, 253–256 (2016)

Berrar, D.: Cross-validation. Encyclopedia Bioinformat Comput Biol **1**, 542–545 (2019)

Capuano, N., Caballé, S.: Multi-attribute categorization of mooc forum posts and applications to conversational agents. In: International Conference On P2p, Parallel, Grid, Cloud And Internet Computing, Springer, 505–514 (2019)

Capuano, N., Caballé, S., Conesa, J., Greco, A.: Attention-based hierarchical recurrent neural networks for mooc forum posts analysis. J. Ambient Intell. Hum. Comput. **12**, 9977–9989 (2021)

Chandrasekaran, M., Ragupathi, K., Kan, M.-Y., Tan, B.: Towards feasible instructor intervention in mooc discussion forums (2015a)

Chandrasekaran, M. K., Kan, M.-Y., Tan, B. C., Ragupathi, K.: Learning instructor intervention from MOOC forums: early results and issues. Arxiv Preprint arXiv:1504.07206 (2015b)

Chaturvedi, S., Goldwasser, D., Daumé Iii, H.: Predicting instructor's intervention in mooc forums. In: Proceedings Of The 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1501–1511 (2014)

Chua, S.-M., Tagg, C., Sharples, M., Rienties, B.: Discussion analytics: identifying conversations and social learners in futurelearn moocs. In: Mooc Analytics: Live Dashboards, Post-Hoc Analytics And The Long-Term Effects**,** 36–62 (2017).

Clavié, B., Gal, K.: Edubert: pretrained deep language models for learning analytics. Arxiv Preprint arXiv:1912.00690 (2019)

Coulombe, C.: Text data augmentation made simple by leveraging Nlp cloud Apis. Arxiv Preprint arXiv:1812.04718 (2018)

Cristea, A. I., Alamri, A., Kayama, M., Stewart, C., Alsheri, M., Shi, L.: Earliest predictor of dropout in moocs: a longitudinal study of futurelearn courses. Association for Information Systems (2018)

Crossley, S., Mcnamara, D. S., Baker, R., Wang, Y., Paquette, L., Barnes, T., Bergner, Y.: Language to completion: success in an educational data mining massive open online class. In: International Educational Data Mining Society (2015)

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. Arxiv Preprint arXiv:1810.04805 (2018)

Durksen, T.L., Chu, M.-W., Ahmad, Z.F., Radil, A.I., Daniels, L.M.: Motivation in a mooc: a probabilistic analysis of online learners' basic psychological needs. Soc. Psychol. Educ. **19**, 241–260 (2016)

Fonseca, S. C., Pereira, F. D., Oliveira, E. H., Oliveira, D. B., Carvalho, L. S., Cristea, A. I.: Automatic subject-based contextualisation of programming assignment lists. International Educational Data Mining Society (2020)

Guo, S.X., Sun, X., Wang, S.X., Gao, Y., Feng, J.: Attention-based character-word hybrid neural networks with semantic and structural information for identifying of urgent posts in mooc discussion forums. IEEE Access **7**, 120522–120532 (2019)

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: review of methods and applications. Expert Syst. Appl. **73**, 220–239 (2017)

Hodgson, R., Cristea, A., Shi, L., Graham, J. Wide-scale automatic analysis of 20 years of its research. In: International Conference On Intelligent Tutoring Systems, Springer, 8–21 (2021)

Jiang, S., Williams, A., Schenke, K., Warschauer, M., O'dowd, D.: Predicting mooc performance with week 1 behavior. In: Educational Data Mining 2014 (2014)

Jordan, K., Goshtasbpour, F.: Jime virtual special collection–2012 To 2022: The decade of the mooc (2022)

Joseph, M.R.: Role of moocs in modern education. J Appl. Sci. Res. **8**, 13–17 (2020)

Jungiewicz, M., Smywiński-Pohl, A.: Data augmentation for sentiment analysis in english–the online approach. In: International Conference on Artificial Neural Networks, Springer, 584–595 (2020)

Khodeir, N.A.: Bi-Gru urgent classification for mooc discussion forums based on bert. IEEE Access **9**, 58243–58255 (2021)

Kobayashi, S.: Contextual augmentation: data augmentation by words with paradigmatic relations. Arxiv Preprint arXiv:1805.06201 (2018)

Lallé, S., Conati, C.: A data-driven student model to provide adaptive support during video watching across moocs. In: International Conference On Artificial Intelligence In Education, Springer, 282–295 (2020)

Li, S., Ao, X., Pan, F., He, Q.: Learning policy scheduling for text augmentation. Neural Netw. **145**, 121–127 (2022)

Liu, P., Wang, X., Xiang, C., Meng, W.: A survey of text data augmentation. In: 2020 International Conference On Computer Communication And Network Security (Ccns), IEEE, 191–195 (2020).

Madabushi, H. T., Kochkina, E., Castelle, M.: Cost-sensitive bert for generalisable sentence classification with imbalanced data. arxiv Preprint arXiv:2003.11563 (2020)

Makcedward: Makcedward/Nlpaug (2020)

Mousavinasab, E., Zarifsanaiey, N.R., NiakanKalhori, S., Rakhshan, M., Keikha, L., GhaziSaeedi, M.: Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. Interact. Learn. Environ. **29**, 142–163 (2021)

Pereira, F. D., Pires, F., Fonseca, S. C., Oliveira, E. H., Carvalho, L. S., Oliveira, D. B. & Cristea, A. I.: Towards a human-ai hybrid system for categorising programming problems. In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, 94–100 (2021).

Qiu, S., Xu, B., Zhang, J., Wang, Y., Shen, X., De Melo, G., Long, C., Li, X.: Easyaug: an automatic textual data augmentation platform for classification tasks. Companion Proc. Web Conf. **2020**, 249–252 (2020)

Raghu, M., Schmidt, E.: A survey of deep learning for scientific discovery. Arxiv Preprint arXiv:2003.11755 (2020)

Rizvi, S., Rienties, B., Rogaten, J., Kizilcec, R.F.: Beyond One-size-fits-all in MOOCS: variation in learning design and persistence of learners in different cultural and socioeconomic contexts. Comput. Hum. Behav. **126**, 106973 (2022)

Rossi, D., Ströele, V., Campos, F., Braga, R., David, J. M. N.: Identifying pedagogical intervention in moocs learning processes: a conversational agent proposal. In: Anais Do Xxxii Simpósio Brasileiro De Informática Na Educação, Sbc, 849–860 (2021)

Shimabukuro, J.: What's wrong with moocs: one-size-fits-all syndrome (2016)

Shorten, C., Khoshgoftaar, T.M., Furht, B.: Text data augmentation for deep learning. Journal of Big Data **8**, 1–34 (2021)

Smaili, E.M., Khoudda, C., Sraidi, S., Azzouzi, S., Charaf, M.E.H.: An innovative approach to prevent learners' dropout from moocs using optimal personalized learning paths: an online learning case study. Stat. Optim. Inf. Comput. **10**, 45–58 (2022)

Stracke, C. M., Bozkurt, A.: Evolution of mooc designs, providers and learners and the related mooc research and publications from 2008 to 2018. In: Proceedings Of International Open & Distance Learning Conference (Iodl19), 13–20 (2019).

Stump, G. S., Deboer, J., Whittinghill, J., Breslow, L.: Development of a framework to classify mooc discussion forum posts: methodology and challenges. In: Nips Workshop On Data Driven Education, 1–20 (2013)

Sun, X., Guo, S., Gao, Y., Zhang, J., Xiao, X., Feng, J.: Identification of urgent posts in mooc discussion forums using an improved Rcnn. In: 2019 IEEE World Conference On Engineering Education (Edunine), IEEE, 1–5 (2019)

Toti, D., Capuano, N., Campos, F., Dantas, M., Neves, F., Caballé, S.: Detection of student engagement in e-learning systems based on semantic analysis and machine learning. In: International Conference On P2p, Parallel, Grid, Cloud And Internet Computing, Springer, 211–223 (2020)

Troyano, J. A., Carrillo, V., Enríquez, F., Galán, F. J.: Named entity recognition through corpus transformation and system combination. In: International Conference On Natural Language Processing (In Spain), Springer, 255–266 (2004)

Wang, W. Y., Yang, D.: That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In: Proceedings of the 2015 Conference On Empirical Methods In Natural Language Processing, 2557–2563 (2015)

Wei, J., Zou, K.: Eda: easy data augmentation techniques for boosting performance on text classification tasks. Arxiv Preprint arXiv:1901.11196 (2019)

Wei, X., Lin, H., Yang, L., Yu, Y.: A convolution-lstm-based deep neural network for cross-domain mooc forum post classification. Information **8**, 92 (2017)

Xiang, R., Chersoni, E., Long, Y., Lu, Q., Huang, C.-R.: Lexical data augmentation for text classification in deep learning. In: Canadian Conference On Artificial Intelligence, Springer, 521–527 (2020)

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., Le, Q. V.: Qanet: combining local convolution with global self-attention for reading comprehension. Arxiv Preprint arXiv:1804.09541 (2018)

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

Laila Alrajhi is a lecturer at King Abdulaziz University's Educational Technology department, Jeddah, SA. She's currently pursuing a Ph.D. in educational data mining, using machine learning and natural language processing to predict the need for urgent instructor intervention in MOOC environments at the Computer Science department, Durham University, Durham, UK. With a master's in Computer Science and a Bachelor's in Computer Science from King Abdulaziz University.

Ahmed Alamri is an assistant professor at Jeddah University's Department of Computer Science. He earned his PhD from Durham University in 2023, specialising in Educational Data Mining. Additionally, he holds a master's degree in Internet and Security and a bachelor's degree in Information and Communication Technology. Ahmed's keen interest lies in educational data mining and learning analytics, utilising machine learning to forecast students' future activities.

Filipe Dwan Pereira received the B.S. degree in Computer Science from Federal University of Roraima, and the Ph.D. degree in Computer Science from Federal University of Amazonas, conducting part of his research at Durham University. Since 2013, he has been an Assitent Professor with the Department of Computer Science, Federal University of Roraima. His area of research covers education data mining, learning analytics, artificial intelligence, machine learning, big data, computing in education, and information systems.

Alexandra I. Cristea is Professor, Deputy Executive Dean of the Faculty of Science, Director of Research and Founder of the Artificial Intelligence in Human Systems research group in the Department of Computer Science at Durham University; Honorary Professor at the Computer Science Department, Warwick University. Her research includes web science, learning analytics, user modelling and personalisation, semantic web, social web, authoring, with over 300 papers on these subjects (over 5700 citations on Google Scholar, h-index 40). She was classified within the top 50 researchers in the world in the area of educational computer-based research according to Microsoft Research. Prof. Cristea has been highly active and has an influential role in international research projects. She has led various projects, has been keynote/invited speaker, organiser, co-organizer, panelist and program committee member of various conferences in her research field. She is a member of the editorial board of the IEEE Transactions on Learning Technologies, executive peer reviewer of the IEEE LTTF Education Technology and Society Journal, Associate Editor of Frontiers in Artificial Intelligence.

Elaine H. T. Oliveira has a Ph.D. in Informatics in Education and a bachelor degree in Computer Science. Since 2002, she has been a Professor in the Institute of Computing at the Federal University of Amazonas, Brazil. She was an associate editor of the Brazilian Journal of Computers in Education (2019-2021), has a Productivity Fellowship in Technological Development and Innovative Extension from CNPq (National Council for Scientific and Technological Development), Brazil, and is a professor member of the Digital Amazonian Girls Program. Her present research is focused on studying students' behavior as they learn how to program, using learning paths and data-driven approaches. The data is collected by the interaction of the students with a self-devised Online Judge. The goals of her research are to predict outcomes, help decision making and provide adaptive learning through Learning Analytics.

Authors and Affiliations

Laila Alrajhi[1,2] · Ahmed Alamri[3] · Filipe Dwan Pereira[4] · Alexandra I. Cristea[1] · Elaine H. T. Oliveira[5]

✉ Laila Alrajhi
laila.m.alrajhi@durham.ac.uk

✉ Alexandra I. Cristea
alexandra.i.cristea@durham.ac.uk

[1] Computer Science, Durham University, Durham, UK

[2] Educational Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[3] Information Systems and Technology, University of Jeddah, Jeddah, Saudi Arabia

[4] Computer Science, Federal University of Roraima, Boa Vista, Brazil

[5] Institute of Computing, Federal University of Amazonas, Manaus, Brazil