Analysing discussion forum data: a replication study avoiding data contamination

Elaine Farrow
School of Informatics
University of Edinburgh
Edinburgh, UK
Elaine.Farrow@ed.ac.uk

Johanna Moore
School of Informatics
University of Edinburgh
Edinburgh, UK
J.Moore@ed.ac.uk

Dragan Gašević
Faculty of Education
Monash University
Clayton 3800, Australia
Dragan.Gasevic@monash.edu

ABSTRACT

The widespread use of online discussion forums in educational settings provides a rich source of data for researchers interested in how collaboration and interaction can foster effective learning. Such online behaviour can be understood through the Community of Inquiry framework, and the cognitive presence construct in particular can be used to characterise the depth of a student's critical engagement with course material. Automated methods have been developed to support this task, but many studies used small data sets, and there have been few replication studies.

In this work, we present findings related to the robustness and generalisability of automated classification methods for detecting cognitive presence in discussion forum transcripts. We closely examined one published state-of-the-art model, comparing different approaches to managing unbalanced classes in the data. By demonstrating how commonly-used data preprocessing practices can lead to over-optimistic results, we contribute to the development of the field so that the results of automated content analysis can be used with confidence.

CCS CONCEPTS

• **Computing methodologies** → **Cross-validation**; *Supervised learning by classification*; • **Applied computing** → **Education**;

KEYWORDS

replication; data contamination; Community of Inquiry; cognitive presence

ACM Reference Format:
Elaine Farrow, Johanna Moore, and Dragan Gašević. 2019. Analysing discussion forum data: a replication study avoiding data contamination . In *The 9th International Learning Analytics & Knowledge Conference (LAK19), March 4–8, 2019, Tempe, AZ, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3303772.3303779

1 INTRODUCTION

Technology use is now a fundamental part of the educational experience for many students, and its importance is widely recognised in the research community. The significance of online discussion forums, where students can interact with one another and with their tutors, is of particular note. Some courses operate fully online, with the discussion forum playing a central role. Forums are also a vital component in blended learning courses, which combine face-to-face instruction with rich online interaction. Nowadays, even traditional face-to-face courses with large class sizes increasingly use text-based forums such as Piazza[1] and online annotation tools like Hypothesis[2] to manage students' questions.

In addition to their primary role in supporting education through interaction and collaboration, discussion forums can also be used to inform research. The messages exchanged in the forum can be exported as a time-stamped record of the discussion. Forum transcripts of this sort encompass social exchanges as well as task-focussed talk and form a rich source of material for researchers interested in studying how participants work together online, and the ways in which effective learning takes place through discussion.

The Community of Inquiry (CoI) framework for online education has emerged as a powerful tool for analysing and developing effective learning experiences [9]. Since its introduction in 2000, the CoI framework has been used in many studies and found to be both useful and robust [15]. It identifies three main elements that are important for a successful educational experience: a social environment conducive to learning (*social presence*), a well-designed course with ongoing facilitation (*teaching presence*), and the student's own cognitive engagement with the subject matter (*cognitive presence*).

While CoI has been well received and widely adopted by researchers, its application in practical educational contexts has been limited because of the difficulty in measuring the three presences in a timely manner. Early work relied on manual coding to identify evidence of the presences in discussion transcripts, using the coding schemes defined in the CoI framework [10]. However, this is time-consuming, expert work, and cannot be deployed in real time. Another approach collected self-reported measures through surveys [2], but these are too intrusive to use for ongoing monitoring.

In response to these limitations, work has been done to develop automated classifiers using features extracted from transcript data – particularly for cognitive presence, the element considered most basic to success [9]. Corich et al. [6] developed an automated content analysis tool and used it to classify forum messages into one of the four levels of cognitive presence. Waters et al. [25] looked at predicting the level of cognitive presence for entire chains of messages, instead of treating messages in isolation – since cognitive presence is expected to develop over time. Kovanović et al. [15] developed a

[1] https://piazza.com
[2] https://web.hypothes.is

LAK19, March 4–8, 2019, Tempe, AZ, USA

ACM ISBN 978-1-4503-6256-6/19/03…$15.00
https://doi.org/10.1145/3303772.3303779

model that is able to identify the level of cognitive presence with 70.3% accuracy, compared to gold-standard human annotation. This represents the state-of-the-art for English-language data. Neto et al. [23] achieved 83% accuracy following the same methodology to analyse a corpus of messages written in Portuguese.

We observe that many of these models were developed on small data sets, and there have been few replication studies [1] – in common with other areas of the learning sciences and psychology [19, 20]. We note that mistakes in data preprocessing can lead to data contamination, where the same data used to develop a model is inadvertently reused while evaluating the model's performance, giving misleading results. Although learning analytics is a relatively new field, we need to think about these issues now. It is particularly important for automated classification techniques to be evaluated rigorously in order to understand how well they are likely to perform on new data. One notable recent development in this area is the MORF platform for replication of studies on MOOCs [8].

This work aims to address these concerns. Our overall goal is to improve the robustness and generalisability of text analysis methods and their use in learning analytics, so that this work can be used with confidence by researchers in the field. This study looks specifically at potential pitfalls affecting automated classification methods for detecting cognitive presence, in order to draw broader recommendations for the field of learning analytics and to contribute to the above over-arching research goal.

Research Question: How do the results from the state-of-the-art model for English-language data [15] compare with a replication study using current best practice for handling unbalanced classes, and for splitting data into training and test sets?

We found that a best-practice replication study was unable to match the published results from Kovanović et al. [15], and concluded that this was likely to be due to over-fitting of that model to its training data. We demonstrated how such over-fitting can be caused by the application of commonly-used data preprocessing techniques, leading to over-optimistic results.

2 BACKGROUND

2.1 The Community of Inquiry model

Garrison et al. [9] introduced a model of Community of Inquiry (CoI) to describe the necessary aspects of an online educational environment. This widely used model has three dimensions, called *presences*: social, teaching, and cognitive. These can be identified in transcripts of online discussions through the presence of particular words and phrases.

In this work, we will focus on the third of the presences, *cognitive presence*, which is further broken down into four levels or phases:

Triggering Event: the initial question that sparks a discussion.

Exploration: the phase of the discussion when many new ideas are being considered.

Integration: the phase where ideas begin to coalesce into a more coherent form as connections are identified.

Resolution: the final phase, where a conclusion has been reached, perhaps in the form of a hypothesis that can be tested.

Although it is desirable for a discussion to progress through all four phases of cognitive presence, not every discussion will do so. There is a natural imbalance, as a single *triggering event* message is expected to lead to multiple messages in the *exploration* and *integration* phases, and a smaller number in the *resolution* phase. Often discussions can become stalled at the *exploration* phase, perhaps due to fear of ideas being rejected [10]. In relatively shorter discussions, it becomes less likely that *resolution* will be reached.

Early work on identifying the CoI presences in online transcripts relied on manual content analysis [9, 10], and self-reporting using a 34 item Likert-scale survey [2]. For content analysis of transcripts, the most appropriate unit of analysis was found to be a single forum message: each message constitutes one conversational 'turn' within a discussion thread. Each message is coded according to the indicators and socio-cognitive processes listed by Garrison et al. [10]. Sometimes a message can show indications of two distinct phases of cognitive presence. The coding scheme indicates that these should be coded with the higher phase [25]. This is sometimes referred to as *coding up*. In addition, some parts of a discussion will not relate to cognitive presence at all; for example, social greetings and expressions of thanks. These can be left unassigned, or given a distinct label, such as *other*.

The CoI framework itself is widely used and has been found to be robust. The transcript-based approach has shown that the coding schemes used to identify the 3 CoI presences, and the 4 phases of cognitive presence in particular, can be applied consistently by different researchers. Only the time and effort required for manual coding has restricted its uptake. For example, the 1747 messages in the data set used in this study took two experienced coders around 130 hours each [11], and they reached agreement in over 98% of cases (Cohen's $\kappa = 0.97$). The prohibitive time cost means that use of the CoI framework has generally been limited to small research projects rather than wider deployment in educational settings.

Clearly, an automated approach to detecting cognitive presence would allow the CoI framework to be used more widely, perhaps even for real-time monitoring. If we can determine the level of cognitive presence demonstrated in each message, we can use this to track the development of the discussion over time. Ideally every discussion would progress through all four phases, from *triggering event* to *resolution*. If a discussion stalls at the *exploration* phase, an instructor may wish to intervene to encourage students to move on to *integration*. In a similar way, instructors could prompt students to move from *integration* to *resolution* after a suitable interval. Meanwhile, a large number of off-topic (*other*) messages could indicate that the discussion forum was not being used as intended. We expect that the thresholds for such interventions would be a matter for the instructor's professional judgement.

2.2 Detecting cognitive presence automatically

Several researchers have proposed methods for automating the coding of cognitive presence. In this subsection, we review some of the methods used and results obtained.

A study using neural networks (NNs) to detect the phases of cognitive presence automatically through content analysis (McKlin [21]) used mainly dictionary-based features, along with 5 features describing the position of the message in the threaded discussion.

Human coders each annotated a sample of the data, and a subset was used to train a neural network. Inter-rater reliability was calculated using the held-out data, both for pairs of human coders, and between the neural network and the consensus value assigned by the humans. The values were found to be comparable, with Cohen's κ in the region of 0.70 and agreement around 81%, indicating that the neural network was able to code the messages with near-human accuracy. However, the *resolution* class was not used in this study, instead being folded into *integration*, as instances were so rare in the data. The most predictive feature was the word count, followed by the number of questions the message contained.

Corich et al. [6] developed a general-purpose automated content analysis tool (ACAT) to eliminate the need for manual segmentation and counting in quantitative content analysis studies, and used it to label cognitive presence on a small data set that had already been coded manually [5]. The tool assigned a class label to every unit, whereas human coders had left some units uncategorised to indicate that there were no traces of cognitive presence at any level. Although the overall distribution of sentences across the classes was similar, the correlation between the two manual coders was much higher (87%) than the correlation between the manual and automated coding (71%). These results are not directly comparable with other studies of cognitive presence because the data was analysed at the sentence level (484 sentences) rather than labelling the 74 messages directly. Insufficient details of the coding scheme were given to allow for replication.

Another exploratory study (Kovanovic et al. [14]) used support vector machines (SVMs) to classify cognitive presence using standard bag-of-words text features. 10-fold cross-validation was used to assess the usefulness of different features. The best model achieved 58.4% accuracy, with Cohen's $\kappa = 0.41$. However, while the feature space was large, with over 20,000 features, the data set used (the same as in the present study) had only 1747 data points. This mismatch in dimensions greatly increases the likelihood of a model over-fitting to the data used for training, rather than learning a general pattern that will also apply to new data. Additionally, none of the features capture anything about the discussion context. This is important, because the definitions of the phases of cognitive presence mean it is very unlikely that a discussion will begin with a *integration* message, or that a *triggering event* message would be followed immediately by a *resolution* message. Thus, adding features relating to the context would be expected to improve the model.

One approach to exploiting the temporal and contextual aspect of discussion threads was explored by Waters et al. [25] using conditional random fields (CRFs). Again, the data set used was the same as in the present study. The forum discussion data was structured as 84 separate threads, each addressing a single topic. The forum interface allowed hierarchically branching discussion: messages with replies, and replies with their own replies. The key feature of the CRF algorithm is that it generates a label *sequence* for an entire sequence of messages, rather than considering messages in isolation. However, it can only handle linear sequences without any branching. Linear 'chains' of messages were therefore extracted from the discussion structure, each chain extending from the 'root' message down to a distinct 'leaf' message. As a result, messages with multiple replies were extracted and analysed repeatedly as part of several chains. 70% of the data was used to train a model, with 20% used for validation and 10% held back for the final test. When the final model was applied to the held-out data, the message chains were analysed and coded separately, and then recombined. A majority vote was used to label each message with a final label. The model achieved 64.2% accuracy, with Cohen's $\kappa = 0.482$.

The current state-of-the-art results for the data used in this present work come from a study by Kovanović et al. [15] using random forests (RFs) [3]. The classifier achieved accuracy of 70.3%, with Cohen's $\kappa = 0.63$ on a held-out portion of the data. The goal was not just to build a high-scoring classifier, but also to gain insight into how cognitive presence is manifested in the discussion through feature analysis. For this reason, the features used in the model were selected because they were theoretically motivated and had potential explanatory power. Around 100 times fewer features were used than in the authors' previous study on the same data (Kovanovic et al. [14]), described above. Word counts derived using the LIWC software package [24], which is based on extensive empirical research, make up 91 of the features. Next, Coh-Metrix [22] was used to generate 106 metrics related to text coherence, complexity, readability, and lexical category use. In addition to these lexical and linguistic features, two further features were defined to represent internal coherence across the sentences within a message, and a count of relevant named entities in the message; along with 6 structural features representing the relative position of the message in the discussion. The data was preprocessed to redress the class imbalance before splitting it to create a training set and a test set (75:25 ratio). Results indicated that deeper levels of cognitive presence were associated with longer messages using more complex language, while *triggering event* messages tended to feature more question marks. These results are similar to McKlin [21]. The number of named entities in a message was also highly predictive, and tended to increase with the level of cognitive presence.

The final study we review adopted the same methodology as Kovanović et al. [15], and applied it to discussion forum data written in Portuguese (Neto et al. [23]). Text analytics tools are not as readily available for other languages as they are for English. The Portuguese version of Coh-Metrix reports 48 measures (compared with 108 for English) and no version of LIWC exists for Portuguese. Adapted versions of 24 of the LIWC features were extracted by the authors, including all those which previously performed well on English-language data. In total, 87 features were used. Stratified sampling was used to obtain the same distribution of cognitive presence levels in the training and test sets, then the training data was preprocessed to redress the class imbalance. The *resolution* class was thereby increased by a factor of 19, from 34 to 646 data points. The test data was not altered. The classifier achieved 83% classification accuracy and Cohen's κ of 0.72 on the held-out test set, higher than the results in Kovanović et al. [15]. The number of question marks was the most predictive feature, followed by average sentence length. Overall, 45% of the top 20 features matched those identified in Kovanović et al. [15].

2.3 Limitations of prior work

One important aspect of prior work where we see room for improvement relates to best practice in training and validating predictive classifiers. We address this in two parts.

2.3.1 Best practice for avoiding over-fitting. Recent work [16] addressed the issue of over-fitting when working with a large number of features but a small data set. In cases like this, there is often a need for parameter selection before building the final model. It is important not to use the held-out test data during this step, as it needs to remain unseen to provide an accurate assessment of the model. In the same way, if the training data will be split into smaller subsets, for example using cross-validation (Section 2.4.2), then only the relevant subsets should be used for parameter selection. A related issue can arise when using over-sampling to address a class imbalance. We look more closely at the details in Section 2.4.4, but note here that both the studies that used SMOTE to redress class imbalance [15, 23] are affected.

2.3.2 Best practice in training and validating models. A review of studies predicting dropout in MOOCs [26] noted that the practice of evaluating models on data sampled from the same course on which they were trained was widespread. In a study that used the same set of features to train several predictive models [26], the results were seen to vary systematically with the evaluation conditions. Using a random split to generate training data and test data from the same course(s) led to accuracy estimates that were significantly over-optimistic compared to testing on a later run of the course. Other work on replication [8] recommends using data from new sessions of the course for validation, rather than taking a random sample from the same course. This practice avoids the possibility of over-fitting to the training data. It also noted that the first run of a course was often unrepresentative of subsequent runs.

None of the studies we reviewed in Section 2.2 validated their models on data from a later run of the course. One reported correlation between automated and human coding on the whole data set [6]. Another [14] reported the results from cross-validation. The rest held back a random sample of the data to use for validation of the chosen model. A summary is shown in Table 1.

2.4 Evaluation methods

2.4.1 Outcome metrics. Accuracy is one of the most commonly used metrics for measuring the power of a predictive classifier: it is simply the proportion of the data points that are classified correctly. When dealing with unbalanced classes, as we are here, two alternative measures are often used which are more informative than accuracy: Cohen's κ, and the macro-averaged F_1 score.

Cohen's κ measures agreement between pairs of annotators on a task involving assigning data points to mutually exclusive categories or classes. It discounts the potential for agreements due to chance and produces a robust estimate of inter-rater agreement, and can thus be used to assess the overall reliability of the coding scheme. Tasks involving human judges usually aim to achieve Cohen's κ scores above 0.70 [17].

Macro-averaged F_1 is found by calculating the F_1 score (the harmonic mean of precision P and recall R) separately for each of the classes in turn, and then taking the average. In this way, good performance across all classes is rewarded more than high performance only on the larger classes. It is an appropriate metric to use when the correct identification of instances of all classes is equally important.

2.4.2 Cross-validation. It is common practice in machine learning to split the data set into a training set, used to build the model, and a held-out test set that is used to estimate how well the model will classify new data. Often, the training data is subdivided further, with some of the data kept back to be used as a validation set to allow comparison of different values for model parameters, before the best model is tested on the held-out data.

When the data set is small, cross-validation can be used instead of creating a validation set. Here, the training data is divided at random into smaller subsets, called *folds*. 10 folds is a common choice. One of these folds is kept back; the model is trained on the rest and evaluated on the final one. This step is repeated until all of the folds have been used for evaluation. The process can be run again using a different random split into folds. The average across the different trials is a good estimate of the expected score for a model created using the whole of the training data.

Cross-validation is useful for comparing different models, but a final evaluation on held-out test data is still needed to estimate how well the best model will perform on completely new data.

2.4.3 Dealing with unbalanced classes. Unbalanced classes often lead to poor classifier results. If a model has seen very few examples of one of the classes during training, it will be harder for it to identify that class in new data. One way to rebalance the classes is to down-sample larger classes by discarding data points until they are closer to the size of the smallest class. However, as the data sets we are considering in this study are already small, down-sampling would discard an unacceptably large proportion. Another approach involves up-sampling the smaller classes, for example by duplicating elements, until the number of instances reaches the size of the largest class. When classes are very unbalanced, as is usual with cognitive presence, such duplication would give us many identical copies of each element in the smallest class.

A better alternative is to create synthetic data points with values that correspond to the distribution of the existing instances in the smaller classes, without being exactly identical to any of them. A popular method is SMOTE: Synthetic Minority Over-sampling TEchnique [4]. For each original data point in turn, SMOTE selects one of its k nearest neighbours and creates a new data point whose values lie between the original point and the neighbour (Figure 1). The value of k can be configured: 5 is a typical default. The process is repeated for the required number of iterations. Thus, the number of new data points created is always a multiple of the size of the original data set. The method proposed by Chawla et al. [4] is only designed for a 2-class problem, but it can be extended to multiple classes using a one-versus-many approach, where each class in turn is contrasted with the largest class.[3]

2.4.4 Class rebalancing and data contamination. In the case where up-sampling uses simple duplication of minority class data points, it is clear that there is a risk of data contamination if the split into training and test sets is not done carefully. If the up-sampling step takes place before the initial split, some of the data points in the test set, which are meant to be unseen data, may actually be duplicates of elements in the training set. This will lead to artificially

[3] Alternative approaches to the problem of learning from unbalanced classes include cluster-based oversampling [12], cost-sensitive learning and algorithmic modification [18], none of which are commonly used in this field nor examined further here.

Table 1: Methods and summary results from prior work (Section 2.2), and the experiments reported in this paper.

	Method	Cross-validation	Class rebalancing	Selection of test data	Outcome metrics		
					Accuracy	Cohen's κ	Macro F_1
McKlin [21]	NN	none	none	random sample (25%)	*n/a*[a]	0.70	*n/a*
Corich et al. [6]	ACAT	none	none	full data set	*n/a*[b]	*n/a*	*n/a*
Kovanovic et al. [14]	SVM	10-fold	none	none	58.4%	0.41	*n/a*
Waters et al. [25]	CRF	none	none	random sample (10%)	64.2%	0.48	*n/a*
Kovanović et al. [15]	RF	10-fold	before split	stratified sample (25%)	70.3%	0.63	*n/a*
Neto et al. [23]	RF	10-fold	after split	stratified sample (25%)	83%	0.72	*n/a*
Experiment 1	RF	10-fold	before split	stratified sample (25%)	70.1%	0.63	0.69
Experiment 2	RF	10-fold	inside CV loop	stratified sample (25%)	61.7%	0.46	0.58
Experiment 3	RF	10-fold	inside CV loop	final course run (13.6%)	54.9%	0.38	0.54

[a] McKlin [21] reported percentage agreement = 81%.
[b] Corich et al. [6] reported Hosti's coefficient of reliability = 0.71.

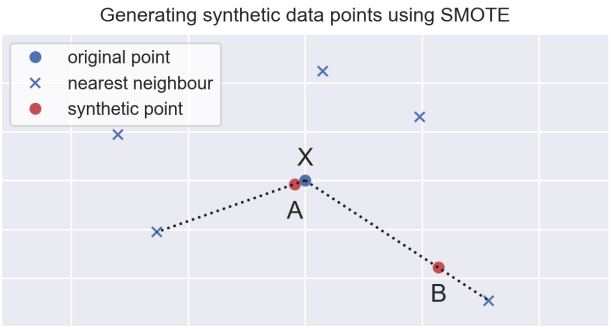

Figure 1: Example of synthetic data point creation in 2 dimensions. Each new point is created somewhere between an existing point X and one of its 5 nearest neighbours. Here, point A is very similar to the original data point, while point B is closer to the chosen neighbour.

high estimates of the power of a model: it is easy for a model to predict the correct values for the test data when those data points have already been seen during training. On genuine held-out data, prediction performance can thus be expected to be worse.

The same fundamental issue arises when synthetic data points are created. If the test data contains synthetic data points that were constructed from elements in the training set, then it is again likely that estimates of the model's power will be artificially high. For example, point *A* in Figure 1 is very similar to the original data point from which it was created. If the original point was part of the training data, and point *A* was in the test data, even a poor model would have a good chance of predicting it correctly.

A more subtle case of data contamination due to over-sampling is explored by Kuncheva and Rodríguez [16] and is highly relevant to the current study. When training classifiers for high-dimensional data with few instances, a common error is to use the same data for selecting features (or tuning model parameters) as for evaluating the final model. Again, this leads to over-optimistic, heavily biased results. Instead, the class rebalancing step needs to be performed for each fold of the cross-validation separately, using only the training data for that fold.[4] Thus, the solution is to do both tuning and model building *inside* the cross-validation loop, so that the same data is used for both steps. Once parameter tuning is complete, a final model can be built using the full training partition as usual.

The effect of this second error is different from the first. Whereas data contamination between the training data and the 'held-out' test data leads to overly positive evaluation results for the final model, contamination introduced at the parameter tuning step can lead to a sub-optimal model being selected.

3 METHODOLOGY

We ran a replication study designed to address our research question. By focussing on one specific data set and classifier type, we can critically examine some of the common pitfalls associated with typical data preparation practices that are in widespread use. We first recreated the state-of-the-art predictive model from Kovanović et al. [15] using the original data and methodology, then applied insights from best practice when working with small data sets [16] to compare different algorithms for dealing with the unbalanced classes in the outcome variable. Building on these results, we explored the effect of splitting the data by course session instead of using a random split, in line with best-practice recommendations for replication studies in an educational context [8].

3.1 Description of the data set

The work presented here makes use of the same data that was used in the Kovanović et al. [15] study that achieved state-of-the-art results, allowing us to compare our results directly. The data was collected from a Masters-level software engineering course that ran at a Canadian university between 2008 and 2011. It was a fully online distance-learning course. The total number of students across all 6 offerings of the course was 85, with a median of 14 students per session (Table 2).

The data consisted of 1747 messages posted on a class discussion forum during weeks 3–5 of each 13-week course offering. Each message was annotated with 205 classification features, as described in Section 2.3. Two expert coders manually annotated all the messages

[4] As less data is available at this stage, it may be better to avoid down-sampling larger classes and simply create new instances of the smaller classes until they match the size of the largest class.

Table 2: Statistics for the 6 offerings of the course.

Session	Student count	Message count
Winter 2008	16	212
Fall 2008	24	633
Spring 2009	12	243
Fall 2009	9	63
Winter 2010	15	359
Winter 2011	13	237
Average (SD)	14.8 (5.1)	291.2 (192.4)
Median	14	240
Total	85	1747

with the level of cognitive presence (98.1% agreement, Cohen's $\kappa = 0.974$).

The discussions were structured around individual video presentations that students recorded and uploaded. Each discussion thread began with a message giving the URL to the video along with information about the research paper being presented. Discussion was conducted asynchronously, and all subsequent messages in the thread were text-only. Participation in the discussion counted for 10% of the final course mark. More details about the assignments and the course structure can be found in [11].

The number of messages varied widely across the levels of cognitive presence (Table 3), as expected. The discussions took place relatively early in the course and were designed to prepare students for their individual research projects, so it is not surprising that relatively few messages indicated that students had reached the *resolution* phase of cognitive presence. A simple baseline classifier assigning the majority class (*exploration*) to every message would achieve 39% accuracy.

Table 3: Breakdown of messages by cognitive presence level.

Cognitive Presence	Count	Percentage
Other	140	8.01%
Triggering Event	308	17.63%
Exploration	684	39.15%
Integration	508	29.08%
Resolution	107	6.12%
All	1747	100.00%

3.2 Methods for managing unbalanced classes

As the number of data points belonging to each outcome class (*i.e.* each phase of cognitive presence and *other*) is highly unbalanced (Table 3) and we know that unbalanced data can cause problems for classification techniques, we used SMOTE (Section 2.4.3) to rebalance the classes in our training data in all our experiments.

We implemented two new R methods for rebalancing our data. These make use of the existing SMOTE implementation in the `DMwR` package for 2-class data and extend it to handle the multi-class case. The first method generates the optimum number of full generations of synthetic points for each of the smaller classes to bring it as close as possible to the size of the original largest class. This means that if the classes are already close in size, no new data points will be created. We call this method *SMOTE multi*, since the number of synthetic data points is always a multiple of the original class size (Figure 2). It is inspired by prior work [23], which used SMOTE to rebalance the training data in this way.

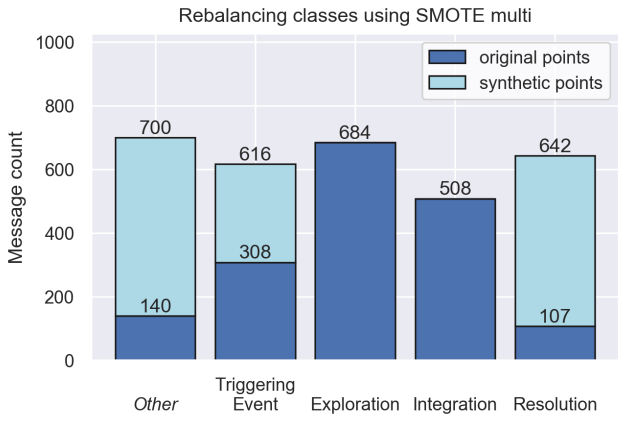

Figure 2: The SMOTE multi algorithm generates synthetic data points in multiples of the original class size, to make the final size as close as possible to the majority class. This example demonstrates its behaviour when applied to the full dataset.

Our second approach starts by creating enough full generations of synthetic data points to match or exceed the target size, then uses random selection within the final generation (only) to make the sizes exactly equal. We call this *SMOTE exact*, because the number of synthetic data points is controlled such that the final class sizes match exactly (Figure 3). This version is closer to the approach taken in Kovanović et al. [15], where all the classes were balanced to be the same size, except that we do not down-sample the larger classes (Figure 4).

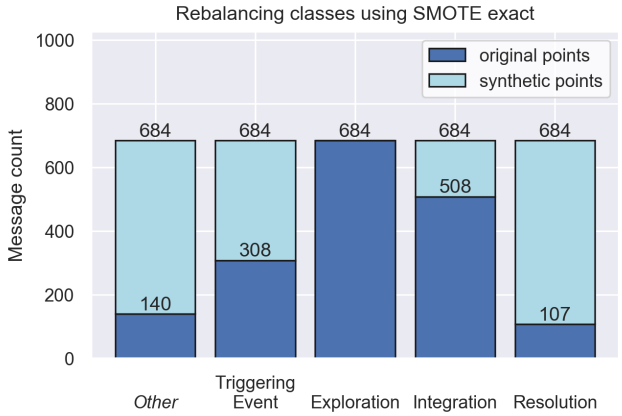

Figure 3: The SMOTE exact algorithm generates the exact number of synthetic data points needed to make every class the same size. By default, it will make all the classes the same size as the largest class.

The SMOTE method as described in Chawla et al. [4] and implemented in R only works correctly with continuous variables. For categorical variables with multiple classes, the recommended approach from Chawla et al. [4] is to take a majority vote from the k nearest neighbours. There are no multi-class categorical variables in our data, and we assume that simple rounding will give a reasonable result for binary variables.

Another consideration is the need to preserve the data type of each field. In particular, if the original field is an integer type (for example, a word count), then after generating the new synthetic data points, we round the values and cast the field to integer again to ensure that the data type remains the same.

The final issue to note with SMOTE and similar methods is that where several fields in a data set are related, there is no simple way to maintain that relationship for the newly created data points. For example, in our data, a message cannot be the last in its thread if it has a non-zero number of replies – but a synthetic data point could conceivably be created with inconsistent values for these features. We do not make any attempt to correct for this type of error.

3.3 Experiment 1: Baseline replication

Our baseline was a direct replication of the state-of-the-art model [15], using the same methodology and the same data: 1747 messages and 205 features. The data was preprocessed to remove the class imbalance, increasing the size of the smaller classes by using SMOTE to create synthetic instances, and down-sampling the larger classes. This produced a new data set of the same size as the original, but with each class equal in size (Figure 4).[5]

The balanced data was then split into training and test partitions in the ratio 75 : 25, using stratified sampling to ensure the same distribution of cognitive presence phases in the two partitions. The training data thus contained 263 examples from each class, while the test data had 87 examples of each. Just as in [15], the training data was used to build a series of 20 random forests of 1000 trees, exploring different settings for the `mtry` parameter that controls the number of features available as candidates at each split point. The specific values to be tested are automatically determined by the `caret` library based on the number of features in the model; here, they were [1, 12, 23, 34, 44, 55, 66, 76, 87, 98, 108, 119, 130, 140, 151, 162, 172, 183, 194, 205]. 10-fold cross-validation, repeated 10 times, was used to select the best performing parameter value. A final random forest model was built using this value on the full training set. The overall accuracy of the final model was then assessed using the test data.

3.4 Experiment 2: Comparing methods for managing unbalanced classes

In Section 2.4.4 we saw how over-sampling can sometimes lead to data contamination, meaning that classifier results are overly-optimistic. Since the data in our baseline replication was split into training and test sets *after* the creation of the additional synthetic data points in the minority classes, this is a real danger.

To address this concern, we split the original unbalanced data into a new training and test set in the ratio 75 : 25, using stratified sampling as before (Table 4). We preprocessed the training set using our two SMOTE variants, SMOTE multi and SMOTE exact, to produce two further training data sets (Figure 5). The class imbalance in the test data set was not corrected, since our goal was to train a classifier that would perform well on future unseen data.

[5]The rebalanced data set from the prior study was made available to us, so we used that directly rather than recreating it.

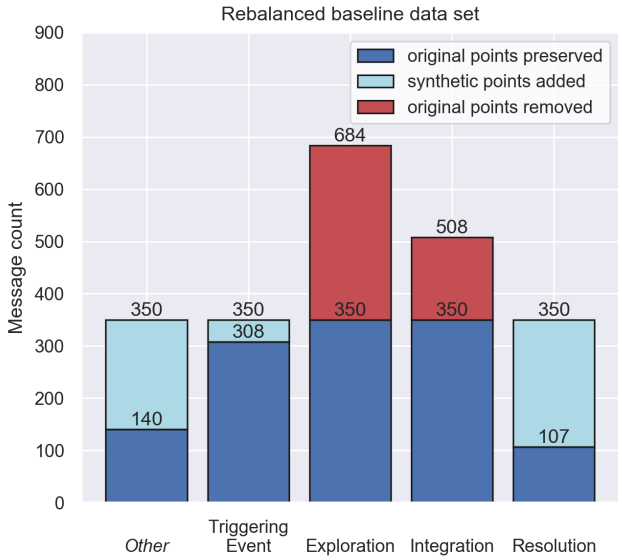

Figure 4: In Kovanović et al. [15], the full data set was rebalanced using SMOTE. Larger classes were down-sampled and smaller classes were up-sampled to create a new data set of the same overall size as the original, but with all classes equally represented.

Table 4: Using a stratified random sample to split the data into training and test sets gave the same distribution of phases of cognitive presence in both cases, matching the distribution in the original data.

	Training		Test	
Cognitive Presence	Count	Percentage	Count	Percentage
Other	105	8.01%	35	8.01%
Triggering Event	231	17.62%	77	17.63%
Exploration	513	39.13%	171	39.15%
Integration	381	29.06%	127	29.08%
Resolution	81	6.18%	26	6.12%
All	1311	100.00%	436	100.00%

We used the unbalanced training data and the two preprocessed training data sets to train three new classifiers using the same procedure as before: 20 random forests of 1000 trees, testing the same set of values for the `mtry` parameter as before, using 10-fold cross-validation, repeated 10 times.

In the same way, we trained two further classifiers that performed the class rebalancing step *inside* the cross-validation loop, following best-practice guidance [16]. We took care to ensure that the same random seed was used to initialise SMOTE every time

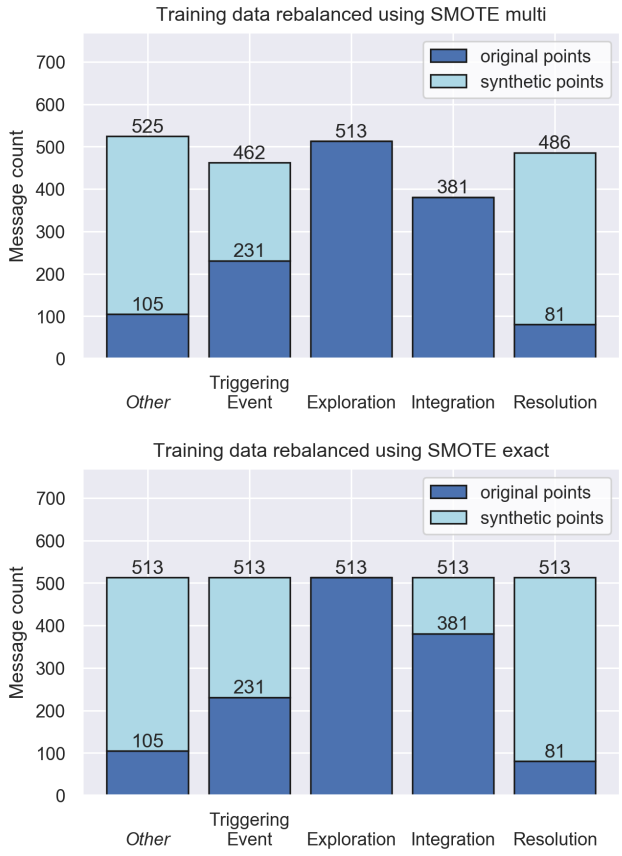

Figure 5: The training data was selected using stratified sampling, then preprocessed to rebalance it using two variants of SMOTE. The class balance in the test data was not altered.

the same cross-validation fold sub-sample was used, to allow a fair comparison of the different values of the `mtry` model parameter. All five models were evaluated on the same held-out test data.

3.5 Experiment 3: Comparing data splits

Using stratified sampling to generate the training-test split allowed us to directly compare our results to prior work [15] when using a different approach to rebalancing the unbalanced classes. However, recent work on replicability of results on MOOCs [8] indicates that it is better to hold out the most recent offering of a course for testing. Even when the instructors are the same, a course changes every time it runs, with the difference between the first and second runs often being the greatest [7]. A useful model needs to be general enough to make predictions on future runs of the course. Therefore, in our last experiment, we used the final course offering (Winter 2011) as test data, and the five earlier offerings for training (Table 5).

We assessed the effect of using this best-practice data split instead of the stratified sample by training and evaluating three classifiers: the first used the unbalanced training data directly, while the other two performed the class rebalancing step inside the cross-validation loop using each of our SMOTE variants. The same procedure was used as in the previous experiments: 20 random forests of 1000 trees, testing the same set of values for the `mtry` parameter, using 10-fold cross-validation, repeated 10 times.

Table 5: Using a session-based split and keeping back the final course offering for testing, the distribution of phases of cognitive presence across the training and test set differed. The test set was not as unbalanced as the training data.

	Training		Test	
Cognitive Presence	Count	Percentage	Count	Percentage
Other	112	7.42%	28	11.81%
Triggering Event	280	18.54%	28	11.81%
Exploration	608	40.26%	76	32.07%
Integration	425	28.15%	83	35.02%
Resolution	85	5.63%	22	9.28%
All	1510	100.00%	237	100.00%

4 RESULTS AND ANALYSIS

4.1 Experiment 1: Baseline replication

As expected, our baseline replication closely approximated the previously reported results (Table 6). We got the same Cohen's κ of 0.63, and an accuracy of 70.1%, rather than 70.3%. The macro-averaged F_1 score was 0.69. The small difference in accuracy is assumed to be due to variations in the random seeds used in the stratified sampling step and the initialisation of the random forest. The parameter tuning step selected 12 as the best value for the `mtry` parameter. The confusion matrix for the baseline replication model is shown in Table 7.

Table 6: Using the procedure from Kovanović et al. [15] and the same rebalanced data set, we closely matched the published results.

Condition	Accuracy	Cohen's κ	Macro F_1
Published baseline	70.3%	0.63	–
Replication of baseline	70.1%	0.63	0.69

Table 7: Confusion matrix for the baseline replication model.

	Predicted				
Actual	*Other*	Triggering	Exploration	Integration	Resolution
Other	**78**	3	3	1	2
Triggering	4	**67**	9	7	0
Exploration	9	15	**36**	27	0
Integration	4	2	22	**44**	15
Resolution	0	0	4	3	**80**

4.2 Experiment 2: Comparing methods for managing unbalanced classes

Using stratified sampling to create the training and evaluation data sets, the classifier that used the unbalanced training data set achieved an accuracy of 60.6%, with Cohen's $\kappa = 0.43$ and macro-averaged $F_1 = 0.52$. Rebalancing the whole training data set before tuning the model, as Neto et al. [23] did, actually decreased the final model's performance compared to using unbalanced data, whereas moving the rebalancing step inside the cross-validation loop improved the results (Table 8).

The best performing model used SMOTE exact inside the cross-validation loop and achieved an accuracy of 61.7%, with Cohen's $\kappa = 0.46$ and macro-averaged $F_1 = 0.58$ (Table 8). The parameter tuning step selected 34 as the best value for the `mtry` parameter. The confusion matrix for this model is shown in Table 9. Compared to the unbalanced case, accuracy was 1.1 percentage points higher, Cohen's κ increased by 0.03, and macro-averaged F_1 by 0.06.

Table 8: Three approaches to managing unbalanced classes, and two variants of the SMOTE algorithm. Data for training and evaluation was selected using stratified random sampling. The best results are in bold.

Condition	Accuracy	Cohen's κ	Macro F_1
No rebalancing	60.6%	0.43	0.52
SMOTE multi preprocessing	58.3%	0.41	0.54
SMOTE exact preprocessing	59.1%	0.42	0.54
SMOTE multi inside the loop	61.2%	0.45	0.57
SMOTE exact inside the loop	**61.7%**	**0.46**	**0.58**

Table 9: Confusion matrix for the best performing model trained on a stratified random sample.

	Predicted				
Actual	*Other*	Triggering	Exploration	Integration	Resolution
Other	**22**	3	8	2	0
Triggering	2	**56**	17	2	0
Exploration	6	17	**110**	36	2
Integration	1	2	43	**75**	6
Resolution	0	0	2	18	**6**

4.3 Experiment 3: Comparing data splits

Using a session-based data split, training on the earlier course sessions and evaluating on the final one, led to much lower results on every metric than when using a stratified random split (Table 10). The best performing model was again the one that used SMOTE exact inside the cross-validation loop. It achieved an accuracy of 54.9%, with Cohen's $\kappa = 0.38$ and macro-averaged $F_1 = 0.54$. The parameter tuning step selected 44 as the best value for the `mtry` parameter. The confusion matrix for this model is shown in Table 11.

Table 10: Splitting the data by course offering and comparing the effect of rebalancing the classes using two variants of the SMOTE algorithm inside the cross-validation loop. The best results are in bold.

Condition	Accuracy	Cohen's κ	Macro F_1
No rebalancing	52.7%	0.33	0.47
SMOTE multi inside the loop	51.9%	0.34	0.52
SMOTE exact inside the loop	**54.9%**	**0.38**	**0.54**

Table 11: Confusion matrix for the best performing model trained using a session-based split.

	Predicted				
Actual	*Other*	Triggering	Exploration	Integration	Resolution
Other	**17**	1	8	2	0
Triggering	1	**23**	4	0	0
Exploration	6	6	**41**	21	2
Integration	2	2	29	**45**	5
Resolution	0	0	5	13	**4**

4.4 Discussion

Despite attempting to closely replicate the prior work from Kovanović et al. [15] and using the same data, we were unable to achieve similar results when we split the data into training and test sets *before* applying class rebalancing: our results were lower on every outcome metric. This leads us to believe that the prior results may have been affected by data contamination between the training and test sets, as we explored in Section 2.4.4, leading to an over-estimation of that model's predictive power.

Experiment 2 demonstrated that carrying out the class rebalancing step *inside* the cross-validation loop led to improvements in classifier performance on the held-out test data, whereas rebalancing the whole training set before tuning the model parameters did not. This is consistent with prior work on parameter tuning with small data sets [16], indicating that tuning inside the cross-validation loop leads to models that generalise better to new data.

The results of Experiment 3, comparing a session-based data split with a stratified random sample, are consistent with recent work on replication in MOOCs. Gardner et al. [8] cite several studies where evaluating predictive models using data from students in the same session of the course resulted in higher estimates of model performance, compared with using data from new sessions of the same course. One explanation for this behaviour is that the data points are not independent. Leaving aside any issues to do with over-sampling, the messages themselves are related to one another and form a natural sequence. They will share commonalities, such as vocabulary used, that go beyond cognitive presence. Taking several messages from a discussion thread to use for training, and then testing the model using another message from the same thread, is likely to give biased results [13]. This also suggests that where splitting by course offering is not possible, discussion threads should be selected for training and testing so that the same participants do not appear in both sections, in order to achieve more reliable

results – although this may not be possible with some very small courses.

We answer our research question by concluding that following a best-practice approach leads to lower results than those reported in Kovanović et al. [15] because of two types of over-fitting. The first is due to the way the class rebalancing step was carried out in this particular study; while the second relates to the widespread practice of using a stratified random sample to split data into training and test sets (Table 1), and means that reported results from other studies may also be over-optimistic.

5 CONCLUSION

Modelling student engagement in online discussion forums is an important topic with the potential to bring benefits to both students and educators. The tools of data science allow us to process large and varied data sets to find patterns in student behaviour and the written content they produce in informal discussion forums that can characterise their engagement with the course. Meanwhile, the Community of Inquiry framework provides a well-developed theoretical grounding for research in this area.

We used a data set of 1747 discussion forum messages to develop several models for predicting a student's level of cognitive presence. With these, we demonstrated the importance of avoiding data contamination in order to build robust classifiers that will generalise well to new data. We illustrated the particular dangers associated with addressing a class imbalance through over-sampling, and showed that rebalancing classes *inside* the cross-validation loop produced a model that achieved the best results on every metric.

It is essential that the data used for the final evaluation is not used to train or tune the model. The best way to ensure this is to split the data by course session and test on the (unseen) final run of the course, rather than following common practice and using a random split – which wrongly assumes that data points from the same session can be treated as independent. As a classifier is only useful to the wider community if it will work well on future runs of a course, we urge the wider adoption of this practice.

ACKNOWLEDGMENTS

Our grateful thanks to Vitomir Kovanović, who generously shared his code. This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

REFERENCES

[1] Juan Miguel L. Andres, Ryan Baker, George Siemens, Dragan Gasevic, and Catherine Spann. 2017. Replicating 21 findings on student success in online learning. *Technology, Instruction, Cognition, and Learning* 10, 4 (2017), 313–333.

[2] J. B. Arbaugh, Martha Cleveland-Innes, Sebastian R. Diaz, D. Randy Garrison, Philip Ice, Jennifer C. Richardson, and Karen P. Swan. 2008. Developing a community of inquiry instrument: Testing a measure of the Community of Inquiry framework using a multi-institutional sample. *Internet and Higher Education* 11, 3-4 (2008), 133–136. https://doi.org/10.1016/j.iheduc.2008.06.003 arXiv:http://ehis.ebscohost.com/

[3] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (jun 2002), 321–357. https://doi.org/10.1613/jair.953 arXiv:1106.1813

[5] Stephen Corich, Kinshuk, and Lynn M Hunt. 2004. Assessing Discussion Forum Participation: In Search of Quality. *International Journal of Instructional Technology & Distance Learning* 1, 12 (2004), 1–12. http://www.itdl.org/Journal/Dec_04/article01.htm

[6] Stephen Corich, Kinshuk, and Lynn M. Hunt. 2006. Measuring critical thinking within discussion forums using a computerised content analysis tool. In *Proceedings of the 5th International Conference on Networked Learning*. 1–8.

[7] Brent J. Evans, Rachel B. Baker, and Thomas S. Dee. 2016. Persistence Patterns in Massive Open Online Courses (MOOCs). *The Journal of Higher Education* 87, 2 (2016), 206–242. https://doi.org/10.1353/jhe.2016.0006

[8] Josh Gardner, Christopher Brooks, Juan Miguel Andres, and Ryan Baker. 2018. Replicating MOOC predictive models at scale. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale - L@S '18*. ACM Press, New York, New York, USA, 1–10. https://doi.org/10.1145/3231644.3231656

[9] D.Randy Garrison, Terry Anderson, and Walter Archer. 2000. Critical Inquiry in a Text-Based Environment: Computer Conferencing in Higher Education. *The Internet and Higher Education* 2, 2-3 (2000), 87–105. https://doi.org/10.1016/S1096-7516(00)00016-6

[10] D. Randy Garrison, Terry Anderson, and Walter Archer. 2001. Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of Distance Education* 15, 1 (2001), 7–23. https://doi.org/10.1080/08923640109527071 arXiv:arXiv:1011.1669v3

[11] Dragan Gašević, Olusola Adesope, Srećko Joksimović, and Vitomir Kovanović. 2015. Externally-facilitated regulation scaffolding and role assignment to develop cognitive presence in asynchronous online discussions. *Internet and Higher Education* 24 (2015), 53–65. https://doi.org/10.1016/j.iheduc.2014.09.006

[12] Haibo He and Edwardo A. Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 9 (2009), 1263–1284. https://doi.org/10.1109/TKDE.2008.239 arXiv:arXiv:1011.1669v3

[13] Sarvnaz Karimi, Jie Yin, and Jiri Baum. 2015. Evaluation Methods for Statistically Dependent Text. *Computational Linguistics* 41, 3 (2015), 539–548. https://doi.org/10.1162/COLI_a_00230

[14] Vitomir Kovanovic, Srecko Joksimovic, Dragan Gasevic, and Marek Hatala. 2014. Automated cognitive presence detection in online discussion transcripts. *CEUR Workshop Proceedings* 1137 (2014).

[15] Vitomir Kovanović, Sreko Joksimović, Zak Waters, Dragan Gašević, Kirsty Kitto, Marek Hatala, and George Siemens. 2016. Towards Automated Content Analysis of Discussion Transcripts: A Cognitive Presence Case. In *LAK '16*. University of Edinburgh, 15–24. https://doi.org/10.1145/2883851.2883950

[16] Ludmila I. Kuncheva and Juan J. Rodríguez. 2018. On feature selection protocols for very low-sample-size data. *Pattern Recognition* 81 (2018), 660–673. https://doi.org/10.1016/j.patcog.2018.03.012

[17] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (1977), 159–174. http://www.jstor.org/stable/2529310

[18] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences* 250 (2013), 113–141. https://doi.org/10.1016/j.ins.2013.07.007

[19] Matthew C. Makel and Jonathan A. Plucker. 2014. Facts Are More Important Than Novelty: Replication in the Education Sciences. *Educational Researcher* 43, 6 (2014), 304–316. https://doi.org/10.3102/0013189X14545513 arXiv:https://doi.org/10.3102/0013189X14545513

[20] Matthew C. Makel, Jonathan A. Plucker, and Boyd Hegarty. 2012. Replications in Psychology Research: How Often Do They Really Occur? *Perspectives on Psychological Science* 7, 6 (2012), 537–542. https://doi.org/10.1177/1745691612460688

[21] Thomas Edward McKlin. 2004. *Analyzing Cognitive Presence in Online Courses Using an Artificial Neural Network*. Ph.D. Dissertation. Georgia State University. http://scholarworks.gsu.edu/msit_diss/1

[22] Danielle S. McNamara, Arthur C. Graesser, Philip M. McCarthy, and Zhiqiang Cai. 2014. *Automated Evaluation of Text and Discourse with Coh-Metrix*. Cambridge University Press, New York. https://doi.org/10.1017/CBO9780511894664

[23] Valter Neto, Vitor Rolim, Rafael Ferreira, Vitomir Kovanović, Dragan Gašević, Rafael Dueire Lins, and Rodrigo Lins. 2018. Automated Analysis of Cognitive Presence in Online Discussions Written in Portuguese. Springer Nature Switzerland AG 2018, 245–261. https://doi.org/10.1007/978-3-319-98572-5_19

[24] Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology* 29, 1 (2010), 24–54. https://doi.org/10.1177/0261927X09351676

[25] Zak Waters, Vitomir Kovanović, Kirsty Kitto, and Dragan Gašević. 2015. Structure Matters: Adoption of Structured Classification Approach in the Context of Cognitive Presence Classification. In *Lecture Notes in Computer Science*, Vol. 9460. 227–238. https://doi.org/10.1007/978-3-319-28940-3_18

[26] Jacob Whitehill, Kiran Mohan, Daniel Seaton, Yigal Rosen, and Dustin Tingley. 2017. MOOC Dropout Prediction: How to Measure Accuracy?. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale - L@S '17*. 161–164. https://doi.org/10.1145/3051457.3053974