Simulation Modelling Practice and Theory 120 (2022) 102616

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Simulation, modelling and classification of wiki contributors: Spotting the good, the bad, and the ugly

Silvia García-Méndez [a,*], Fátima Leal [b], Benedita Malheiro [d,e], Juan Carlos Burguillo-Rial [a], Bruno Veloso [c,f,e], Adriana E. Chis [g], Horacio González–Vélez [g]

[a] Information Technologies Group, atlanTTic, University of Vigo, School of Telecommunications Engineering, Campus Lagoas-Marcosende, Vigo, 36310, Galicia, Spain
[b] REMIT, Universidade Portucalense, Porto, Portugal
[c] Universidade Portucalense, Porto, Portugal
[d] ISEP/IPP, School of Engineering, Polytechnic Institute of Porto, Porto, Portugal
[e] INESC TEC, Porto, Portugal
[f] Faculty of Economics, University of Porto, Portugal
[g] Cloud Competency Centre, School of Computing, National College of Ireland, Dublin, Ireland

ARTICLE INFO

Keywords:
Classification
Data reliability
Stream processing
Synthetic data
Data fabrication
Wiki contributors

ABSTRACT

Data crowdsourcing is a data acquisition process where groups of voluntary contributors feed platforms with highly relevant data ranging from news, comments, and media to knowledge and classifications. It typically processes user-generated data streams to provide and refine popular services such as wikis, collaborative maps, e-commerce sites, and social networks. Nevertheless, this *modus operandi* raises severe concerns regarding ill-intentioned data manipulation in adversarial environments. This paper presents a simulation, modelling, and classification approach to automatically identify human and non-human (bots) as well as benign and malign contributors by using data fabrication to balance classes within experimental data sets, data stream modelling to build and update contributor profiles and, finally, autonomic data stream classification. By employing WikiVoyage – a free worldwide wiki travel guide open to contribution from the general public – as a testbed, our approach proves to significantly boost the confidence and quality of the classifier by using a class-balanced data stream, comprising both real and synthetic data. Our empirical results show that the proposed method distinguishes between benign and malign bots as well as human contributors with a classification accuracy of up to 92 %.

1. Introduction

Crowdsourcing platforms primarily rely on data produced and shared by an extensive network of contributors, known as "The Crowd". Given the ubiquitous Internet access and the increasing user-generated data sources, the number of crowdsourcing systems has dramatically increased in recent times, significantly modifying the online behaviour of users and businesses. People no longer rely on expert advice but on the experiences of other users to make decisions on purchases, leisure, news, learning, and information in general.

* Corresponding author.
E-mail addresses: sgarcia@gti.uvigo.es (S. García-Méndez), fatimal@upt.pt (F. Leal), mbm@isep.ipp.pt (B. Malheiro), J.C.Burguillo@uvigo.es (J.C. Burguillo-Rial), brunov@upt.pt (B. Veloso), adriana.chis@ncirl.ie (A.E. Chis), horacio@ncirl.ie (H. González–Vélez).

https://doi.org/10.1016/j.simpat.2022.102616
Received 9 December 2021; Received in revised form 24 May 2022; Accepted 13 June 2022
Available online 18 June 2022

Such platforms integrate Artificial Intelligence (AI) techniques such as data mining to swiftly process the crowd feedback and provide tailored up-to-date services on the fly using data streams. However, the processing opacity and the voluntary nature of crowdsourced data raise algorithmic transparency and data reliability concerns. On the one hand, the opacity of AI algorithms affects the interpretability of the results and can only be tackled by developers at the algorithmic design stage. On the other hand, data manipulation performed on behalf of third-party interests has become omnipresent in crowdsourcing platforms with fake news in social networks, biased feedback in evaluation-based platforms, and undesired (spam) content in wiki pages. Such underground activities, which negatively impact the reliability of the results, are typically performed at large scale by ill-intentioned bots in adversarial contexts.

Commonly construed as self-activated software agents programmed to run continuously in distributed network environments, bots perform tasks and make decisions on behalf of their creators without direct human intervention. They typically sense, perceive, and adapt to the context they operate. According to Tsvetkova et al. (2017) [1], all four main types of bots – content extractors, action executors, content generators, and human emulators – include benign and malign sorts. While benign bots tend to perform repetitive tasks to improve quality of service and system conditions, malign bots are designed to degrade content and system conditions by conveying false or biased information, altering system parameters, and/or tampering with data sources.

This work addresses the real-time profiling and classification of Wikivoyage contributors into human and non-human (bots) as well as benign and malign contributors. The proposed method relies on data simulation and modelling of wiki streams to obtain an automatic classification of contributors. While the data fabrication balances classes to improve the consistency of the results, data stream modelling builds contributor profiles on the fly.

Therefore, this paper furnishes a data stream classification to identify deceitful sources in real-time. Specifically, contributors are modelled using review-based information (*e.g.*, number of reviews, frequency of reverts, links, and number of characters inserted and deleted) and ORES edit quality,[1] an Application Programming Interface (API) designed for wiki platforms, which helps to automate vandalism detection and removal. The model is incrementally updated with each incoming review. The classifier uses the contributor profile, first, to differentiate humans from bot users. Then, we employ a stack-based Machine Learning (ML) model, *i.e.*, a two-level stacking system, to detect both bot/human users and positive/negative contributions.

The experiments have been conducted using a real Wikivoyage data set containing 417 bots and 46,952 humans. Despite the imbalanced class distribution, the final two-level stacking classifier presents results between 80 % and 95 % concerning the accuracy and F-measure (both macro and micro results per class). To balance classes within the experimental data set, this work adopts a data fabrication scenario based on synthetic-generated data. The real and synthetic data have been combined, improving the system performance and proving the solution efficiency. In summary, our proposal allows us to understand the behaviour of both benevolent and malevolent bots to pre-empt malign contributions and, ultimately, to prevent attacks to wiki pages.

The rest of this paper is organised as follows. Section 2 includes relevant related work concerning contributor modelling and classification. Section 3 introduces the proposed approach, describing the profiling, classification, and data fabrication model. Section 4 presents the empirical evaluation results of the developed classifiers, including the simulation of synthetic data to balance the classes within the experimental data set. Finally, Section 5 concludes and highlights new perspectives to detect ill-intentioned contributors.

2. Related work

Data crowdsourcing has been explored in multiple domains, ranging from knowledge sharing, collaborative maps, social interaction to personalised recommendation [2]. For example, Wikipedia[2] and Wikivoyage[3] offer useful knowledge, which is continuously updated and improved by an invisible army of volunteer contributors; Waze[4] assists drivers based on information provided in real-time by the crowd concerning traffic, accidents, obstacles, and road conditions; Facebook[5] provides a way for people to communicate and socialise based exclusively on peer input; and TripAdvisor[6] makes personalised recommendations relying solely on opinions and classifications shared by the crowd in the form of ratings, textual reviews, or photos. In this continuously evolving data crowdsourcing world, detecting automated data manipulators (bots and spammers) and their intentions (benign or malign) is essential to ensure data reliability. Detection methods search for hyperactivity indicators such as the number of page views, bounce rates, fake conversions, abnormal traffic spikes, and atypical average session duration.[7] The accurate identification and classification of bots and sockpuppets can arguably help to mitigate the effects of malign data manipulation.

[1] Available at www.mediawiki.org/wiki/ORES, December 2021.
[2] Available at www.wikipedia.org, December 2021.
[3] Available at www.wikivoyage.org, December 2021.
[4] Available at www.waze.com, December 2021.
[5] Available at www.facebook.com, December 2021.
[6] Available at www.tripadvisor.com, December 2021.
[7] Available at www.netacea.com/glossary/detect-bot-traffic, December 2021.

2.1. Classification of contributors

Wiki contributors can be humans or bots, and their contributions may be benign or malign [3]. Bot activities are not all flagged and can be mistaken as human contributions. For example, Steiner et al. (2014) [4] identify wiki bots in real-time by checking the bot flag and the name "bot". The authors found in Wikipedia a linear relationship between the number of bots and the number of edits. The most active bots display a classic hockey stick activity curve with a long tail composed of slower bots responsible for corrective repetitive tasks.

Malign activities or vandalism correspond, according to Wikimedia,[8] to edits made with deliberate malign intent, which differ from inadvertently damaging edits. To address Wiki vandalism detection, Adler et al. (2011) [5] propose a trust-based approach called WikiTrust.

Since 2015, Wikimedia has provided the Objective Revision Evaluation Service (ORES[9]) to automate vandalism detection and removal. This ML system predicts the quality of edits (quality of edit probability score) as they are made, as well as the quality of article drafts (quality of article probability score.[10]) The article quality model bases its predictions on structural characteristics of the article, such as the number of sections, references or infobox presence, since these features seem to correlate strongly with good writing and tone. ORES accepts single or batch-edit submissions. More specifically, given an edit, ORES predicts three probabilities: (*i*) whether or not an edit causes damage; (*ii*) if it was saved in good faith; and (*iii*) if the edit will eventually be reverted. Regarding the quality of an article draft, ORES returns the probability distribution of being in one of the predefined classes (spam, vandalism, attack, or OK). These scores can also be used to help identify malign bots. Curiously, Tsvetkova et al. (2017) [1] concluded that even benign Wikipedia bots, which support the encyclopedia in tasks such as correcting spelling, maintaining links, or undoing digital vandalism on pages, often undo each other's edits. Works such as the one presented by Yang et al. (2017) [6] use the power of the crowd to detect malign users. Their solution encompasses a mechanism to encourage users to report collaboratively malign behaviours.

Indeed, malign behaviours in social networks have been explored by many authors in the literature [7], some of them following an ML approach [8]. Notably, Yamak et al. (2016) [9] focus on the detection of multiple online accounts (sockpuppet) created for purposes of deception on the English Wikipedia. The authors select the number of user contributions by namespace; the frequency of revert after each contribution; the average of bytes added and removed from each revision; the average contribution length in the same article; and the interval between user registration and user first contribution. Finally, the best results were obtained with the Random Forest algorithm [10].

When it comes to the popular Twitter[11] platform, it is far from safe from this type of attack. In fact, Velayutham et al. (2017) [11] seek to detect such bots in Twitter using an ML-based approach to distinguish them from human profiles. Consequently, they identified the features related to user profile and content analysis used to compute *botScore*. A similar work by Efthimion et al. (2018) [12] goes beyond identifying bot users through ML and focuses on determining their prevalence in the platform. Lately, the widely used Botometer diagnostic tool[12] has been under study. In this vein, Rauchfleisch et al. (2020) [13] analysed its diagnostic ability over time, concluding that its thresholds are prone to variance with the consequent increase in false positives and false negatives.

Kumar et al. (2016) [14] seek to automatically identify fake articles in Wikipedia. They exploit not only textual content but also temporal data and network features. The authors report that the same group speedily writes insightful conclusions such as hoax articles of relatively new accounts with highly dense or overlapping networks. Conversely, Green & Spezzano (2017) [15] address the problem of identifying spam users on Wikipedia as a binary classification task based on user editing behaviour. They use as features the edit size (average size of edits, standard deviation of edit sizes, variance significance); edit time (average time between edits, the standard deviation of time between edits); links in edit (unique link ratio, link ratio in edits); talk page edit ratio and username. Moreover, they combine these features with the ORES scores to improve the accuracy of the classification. Finally, Heindorf et al. (2016, 2019) [16,17] adopt an online learning approach to the task of wiki vandalism detection.

More closely related to our work, Sarabani et al. (2017) [18] use a Random Forest implementation to classify vandal contributions in Wikidata with 89 % recall and reducing by up to 98 % the contributions which need to be reviewed. This work moved from text-based feature engineering and paid special attention to contextual features and user profiling. Finally, consideration should also be given to hybrid approaches such as the one presented by Zheng et al. (2019) [19]. Mainly, they created a 9-level bot taxonomy based on their behaviour in the English Wikipedia using both unsupervised rules and labelled data.

Concerning map-based crowdsourcing applications, Sanchez et al. (2018) [20] present a method to detect malign behaviours in Waze. The authors explore the Sybil attack, which uses a coordinated group to provide false information. The user behaviour is modelled using graphs where the weight of edges is computed combining distance, time, speed and number of votes.

More sophisticated approaches are also present in the literature. Hall et al. (2018) [21] adopt a gradient boosting classifier to detect previously unidentified bots in wikis based on implicit behavioural and other informal editing characteristics. The adopted model has 15 activity pattern features and 14 revision comment-based features. The latter set of features proved to be highly effective at predicting bot edits.

[8] Available at www.wikimedia.org, December 2021.
[9] Available at https://ores.wikimedia.org, December 2021.
[10] Available at www.mediawiki.org/wiki/ORES, December 2021.
[11] Available at www.twitter.com, December 2021.
[12] Available at https://botometer.osome.iu.edu, December 2021.

Table 1
Comparison of contributor classification approaches.

Approach	Contributor classification	Contribution classification	Synthetic data	Wiki streams
Adler et al. (2011) [5]		✓		
Choi et al. (2016) [8]		✓		
Efthimion et al. (2018) [12]	✓			
Green & Spezzano (2017) [15]	✓			
Hall et al. (2018) [21]	✓			
Heindorf et al. (2016, 2019) [16,17]		✓		✓
Kumar et al. (2016) [14]	✓			
Nikesh et al. (2020) [23]	✓			
Rauchfleisch & Kaiser (2020) [13]	✓			
Sanchez et al. (2018) [20]		✓		
Sarabani et al. (2017) [18]		✓		
Steiner et al. (2014) [4]	✓			
Subrahmanian et al. (2016) [7]	✓			
Tsvetkova et al. (2017) [1]	✓	✓		
Velayutham et al. (2017) [11]	✓			
Yamak et al. (2016) [9]	✓			
Yang et al. (2017) [6]	✓	✓		
Zheng, Albano et al. (2019) [19]	✓			
Zheng, Yuan et al. (2019) [22]	✓	✓		
Our proposal	✓	✓	✓	✓

Furthermore, Zheng et al. (2019) [22] called attention to the need for sufficient training data to detect malign contributors in collaborative communities like wikis. They solved this issue by exploiting a Long Short-Term Memory auto-encoder trained solely with benign users. Then, based on the representations learned, the system uses a Generative Adversarial Network as a discriminator to spot malign contributors.

More recent works, such as Nikesh et al. (2020) [23], address the problem of identifying Wikipedia articles with undisclosed paid content and their contributors. Once again, the authors rely on content-derived features, user, and edit history patterns to build an unsupervised ML system. The authors create a graph where the nodes represent articles together with their PageRank and the edges represent contributors who have edited both articles. This article network has been used to obtain the article local clustering coefficient and PageRank. Specifically, they collect article-based features (age of user account at article creation, presence of infobox, number of references, photos, article categories, and content length); article network-based features (article PageRank, and local clustering coefficient); user-based features (username-based features, average size of added text, average time difference, ten-byte long edit ratio, and percentage of edits on User or Talk pages). Once again, the best results have been obtained with a Random Forest classifier and the selected features combined with ORES article-based features. Finally, it is also worth mentioning the broad survey on fake news detection by Zhang & Ghorbani (2020) [24] where they analyse the negative impact of misinformation on social media and distinguish between human and non-human (bots and cyborgs) creators.

2.2. Research contribution

This article analyses wiki-based contributions adopting data stream simulation, modelling and classification. Data stream modelling builds and continuously updates contributor profiles using the stream of contributions. The proposed profiling model employs the features provided by the surveyed works, *i.e.*, review-based information (*e.g.*, number of reviews, reverts, links, number of characters inserted or deleted, *etc.*) and ORES edit quality. Based on the statistical metrics of real data, our data fabrication produces a class-balanced data set that helps to improve the consistency of the results. Finally, the data stream classification identifies both human and non-human contributors as well as benevolent and malevolent contributors.

Table 1 depicts a comparison of relevant related work that addresses contributor classification. The literature presents a significant number of approaches to classify contributors. However, they do not furnish the contribution type classification, *i.e.*, detection of malevolent and benevolent contributors. Furthermore, the modelling of wiki streams for the online automatic contributor classification has not been thoroughly explored. To overcome this gap, this work addresses contributor profiling and classification to identify malign behaviours in wiki-based crowdsourcing platforms on the fly.

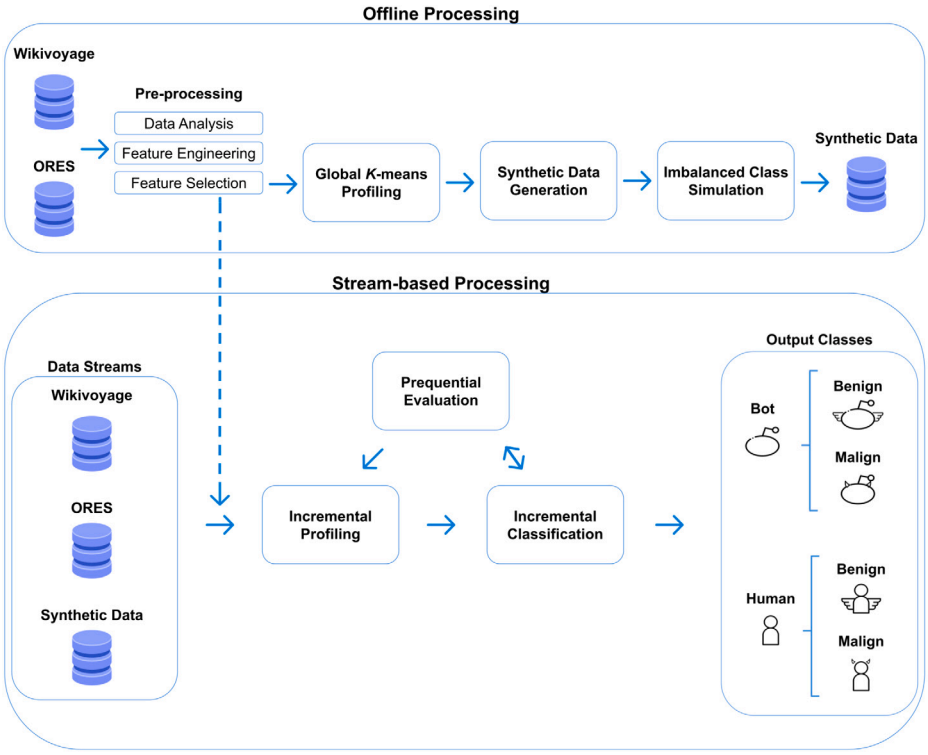

Fig. 1. Proposed stream-based classification of wiki contributors.

3. Proposed method

The proposed method is composed of multiple stages: (*i*) pre-processing; (*ii*) synthetic data generation; (*iii*) classification; and (*iv*) evaluation. Fig. 1 illustrates the solution designed to process wiki-based streams and generate classifications. The pre-processing stage (Section 3.1) analyses the input data in terms of pairwise correlations (Section 3.1.1) and employs feature engineering (Section 3.1.2) to select the best features for the contributor profiling (Section 3.1.3). The synthetic-generated data enables to balance the classes and improve the model effectiveness (Section 3.2). The resulting data set, composed of real and synthetic data, is then incrementally processed to model (Section 3.3) and classify (Section 3.4) the contributors. The classification, which relies on well-known algorithms, explores single-class (binary) as well as a novel multi-class (stacking) classification of contributors and their editions. Finally, the outcomes are evaluated through standard classification metrics (Section 3.5).

3.1. Pre-processing

Pre-processing aims to obtain a consistent feature space and facilitate compatibility with the target classification. This stage assembles three tasks: (*i*) data analysis; (*ii*) feature engineering; and (*iii*) feature selection.

3.1.1. Data analysis

The data analysis starts with a pairwise correlation of the features enumerated in Table 2, which are based on the literature review. Eq. (1) describes the Pearson Correlation Coefficient [25] used to calculate the correlations where x and y represent two different features. The correlation coefficient ranges from −1 to 1. When the relation between two features is inverse, the metric is negative, otherwise it is positive.

$$r_{xy} = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2}\sqrt{\sum(y_i - \overline{y})^2}} \tag{1}$$

3.1.2. Feature engineering

The Wikivoyage data were compiled by the authors between January 14th and June 21st 2020. Wikivoyage holds crowdsourced information related to cities, attractions, monuments, hotels, restaurants, cultural details, *etc*. In particular, the retrieved data set includes 3369 pages, 47,369 contributors, and 319,856 reviews.

Table 2 contains the manufactured features of this experimental data set. The first target corresponds to feature #2, where zero represents human and one represents non-human (bot) contributors. Furthermore, the second target feature, named *contribution type*,

Table 2
Features selected for the classification of contributors and their contributions.

# Num.	Feature name
1	User ID
2	Bot flag
3	Number of reviews
4	Average length of reviews
5	Number of pages
6	Average number of revisions per page
7	Frequency of revisions per week
8	Number of pages revised per week
9	Number of reverts of the contributor reviews
10	Revert frequency of the contributor reviews
11	Links ratio of the reviews
12	Repeated links ratio of the reviews
13	Amount of inserted characters
14	Amount of deleted characters
15	ORES edit quality: true/false damaging & good faith probability
16	Average item quality from ORES: A/B/C/D/E probability
17	Average article quality from ORES: OK/attack/spam/vandalism probability
18	Average article quality from ORES (wp10): B/C/FA/GA/start/stub probability

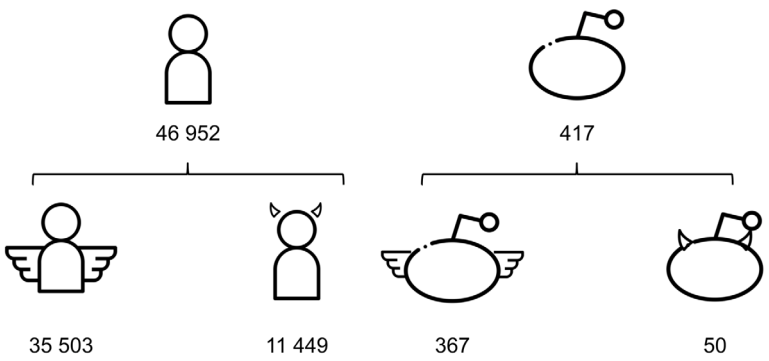

Fig. 2. Distribution of the data set in terms of *user type* and *contribution type* targets.

is derived from feature #17 (OK probability) provided by ORES. When this probability is above 50 %, the contribution is positive or favourable, otherwise, negative or hostile. In particular, zero corresponds to positive and one to negative edits.

Fig. 2 depicts the distribution of contributor classes within the original data set: (*i*) 35,503 benign and 11,449 malign humans; and (*ii*) 367 benign and 50 malign bots.

3.1.3. Feature selection

Feature selection reduces the original feature space to improve performance and to avoid creating computationally unfeasible models. This work, due to the number of input features, adopted Recursive Feature Elimination (RFE,[13]) a wrapper-type feature selection algorithm, to identify the features that maximise performance. RFE was configured to wrap the Linear Support Vector Classifier[14] and use as parameters `penalty=l1`, `dual=False`, `step=0.05` and `n_jobs=-1`.

3.2. Synthetic data generation

Synthetic data generation enables the simulation of stochastic and multi-spectral layouts to enhance ML model testing. This common practice found in the literature [26–29] supports the generation of relevant scenarios that, in most cases, are absent in the real data. Furthermore, it produces anonymous data, thereby maintaining the privacy of information. This process balances experimental data sets with synthetic samples which, in turn, are used to build models that react and operate correctly under the considered scenario.

In classification, it is desirable to have data equally distributed among classes, *i.e.*, ensure the data is free of any class bias. The main benefit of this scenario, where classifier models are built with balanced experimental data, is that they display higher classification accuracy.

[13] Available at http://www.scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html, December 2021.
[14] Available at http://www.scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html, December 2021.

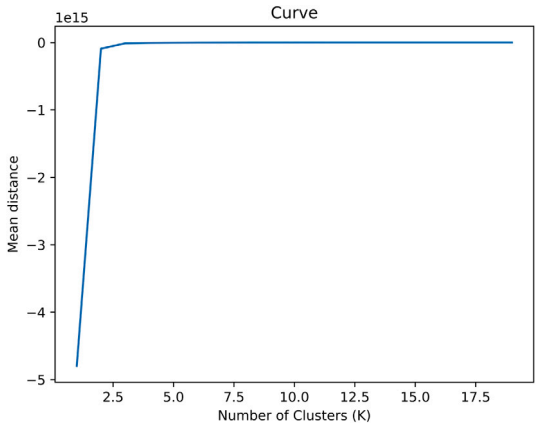

Fig. 3. K-means optimal K value based on mean distance deviation.

In this work, synthetic data generation constitutes a user-controlled, accurate, cost-effective, efficient and scalable way to balance the classes within the experimental data set.

Synthetic samples were created using a statistical approach based on the quartile distribution of the most relevant features for both targets. In particular, the proposed synthetic data generation method creates feasible editor daily incremental activity samples corresponding to the two target categories. The generated samples are composed of the features in Table 2 and based on statistical measures (quartile distribution, median, minimum and maximum values) considering four intervals: (*i*) from minimum to the first quartile (Q1); (*ii*) from Q1 to median; (*iii*) from median to third quartile (Q3); (*iv*) from Q3 to the maximum.

A non-hierarchical K-means cluster analysis[15] was performed on the original data set to identify common bot behaviour. Based on the information shown in Fig. 3, the model was configured to use two clusters ($K = 2$) and trained with bot samples.

K-means provides the required statistic indicators to generate the new data, *i.e.*, quartile distribution, minimum and maximum feature values. Algorithm 1 describes the implemented synthetic data model. It generates 46,532 new samples to balance the class distribution of the original data set. As previously mentioned, the model uses four different groups to create random feature values: (*i*) from minimum to Quartile 1 (Q1); (*ii*) from Q1 to median; (*iii*) from median to Q3; (*iv*) from Q3 to maximum.

Algorithm 1: Synthetic data model

Input: K-means statistic indicators
Data: min, $Q1$, $median$, $Q3$, max
Result: Creates new samples for the experimental data set by generating feature values based on quartile distributions
begin
    $count = 46532$; // Number of samples to be generated
    $ranges = \{min, Q1, median, Q3, max\}$; // Quartile distribution
    $synthetic\_data = []$; // Synthetic samples
    foreach $r \in ranges - 1$ **do**
        foreach $i \in count/4$ **do**
            $entry = random(ranges[r], ranges[r+1])$
            $synthetic\_data.append(entry)$
// Returns the synthetic generated data set
return $synthetic\_data$

Finally, the original and synthetic data are joined into the combined data set. The editions, characterised by the features in Table 2, are aggregated on a daily basis. The resulting data set is then processed as a stream.

3.3. Incremental profiling

The incremental profiling builds and continuously updates the profiles of contributors from the stream of samples stored in the combined data set. The profile of each contributor holds: (*i*) the incremental sum of features #3, #5, #9 and #11-14; (*ii*) the incremental average of features #4, #6, and #15-18; and (*iii*) the manufactured features #7, #8 and #10. The latter are calculated as follows:

[15] Available at http://www.scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html, December 2021.

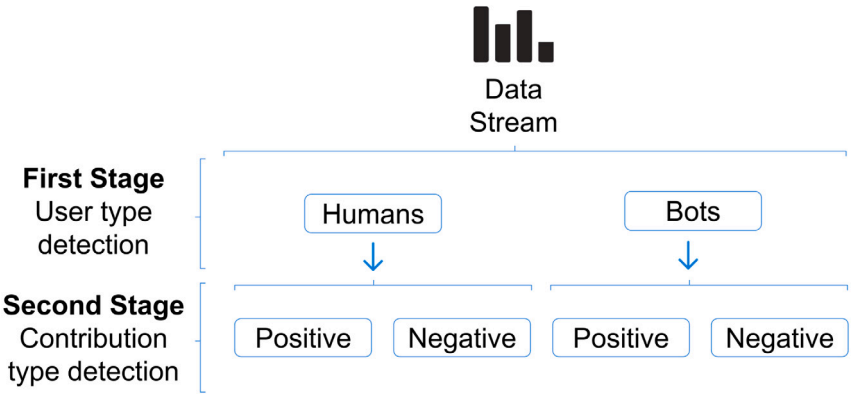

Fig. 4. Proposed data stream multi-class stacked classification of contributors.

- #7 = #3/number of weeks, incremental sum.
- #8 = #5/number of weeks, incremental sum.
- #10 = #9 / #3, incremental sum.

3.4. Classification

The classification task explores single-class (binary) and a novel multi-class (stacking) methods. Several stream-based binary classification algorithms were selected according to performance in similar problems [17,30,31] and availability in scikit-multiflow,[16] the adopted ML package for streaming data. They include single and ensemble methods:

- Naive Bayes (NB)[17]
- Decision Tree (DT)[18]
- Random Forest (RF)[19]
- Boosting Classifier (BC)[20]

The proposed two-level stacking method, depicted in Fig. 4, employs three RF models with the following hyperparameter configuration: `random_state=0`, `n_estimators=15`, `max_features=None`.

3.5. Evaluation metrics

The evaluation of the single-class (binary) and multi-class (stacking) classifiers uses classification accuracy, *F*-measure, macro-averaging and micro-averaging. The classification accuracy determines the classifier performance by measuring the number of correct classifications. The *F*-measure combines Precision and Recall to establish the effectiveness of the classifier. The micro-averaging and macro-averaging methods return a single value for the different metrics across multiple classes. While the macro-average computes the metric average independently for each class, *i.e.*, treats all classes equally; the micro-average aggregates the contributions of all classes to compute the metric average. As suggested in the literature [32,33], the averaging methods should be used when the experimental data set is class-imbalanced.

4. Experimental results

All experiments were performed on a server with the following hardware specifications:

- Operating System: Ubuntu 18.04.2 LTS 64 bits
- Processor: Intel@Core i9-9900K 3.60 GHz
- RAM: 32 GB DDR4
- Disk: 500 GB (7200 rpm SATA) + 256 GB SSD

[16] Available at https://scikit-multiflow.github.io, December 2021.
[17] Available at https://scikit-multiflow.readthedocs.io/en/latest/api/generated/skmultiflow.bayes.NaiveBayes.html, December 2021.
[18] Available at https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.trees.ExtremelyFastDecisionTreeClassifier.html, December 2021.
[19] Available at https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.meta.AdaptiveRandomForestClassifier.html#skmultiflow.meta.AdaptiveRandomForestClassifier, December 2021.
[20] Available at https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.meta.OnlineBoostingClassifier.html, December 2021.

Table 3
Most correlated features with *user type* (1) and *contribution type* (2) targets.

Target	Correlated features and correlation value									
1	#5 0.16		#7 0.55		#8 0.57		#13 0.16		#14 0.22	
2	#3 −0.24	#5 −0.24	#9 −0.18	#13 −0.20	#14 −0.17	#15[a] −0.16	#15[b] 0.16	#18[c] −0.18	#18[d] −0.17	#18[e] −0.19

[a]Damaging false probability.
[b]Damaging true probability.
[c]B probability.
[d]C probability.
[e]FA probability.

The experiments encompass: (*i*) offline feature analysis to select the most relevant features to the contributor profile; (*ii*) offline synthetic data generation to balance the classes within the data set; and (*iii*) classification of human and bot contributors as well as benign and malign contributions, *i.e.*, regarding the two target features. Moreover, the classification stage explores three scenarios: (*a*) stream-based binary classification of human and bot contributors using imbalanced real data; (*b*) stream-based binary classification of benign and malign contributors using imbalanced real data; and (*c*) stream-based multi-class stacked classification of benign and malign humans and bots using imbalanced (real) and balanced (real and synthetic) data. While scenarios (*a*) and (*b*) set the classification baseline with the original deeply unbalanced experimental data for each of the two target features, scenario (*c*) analyses the performance of the stacking classification strategy in the simultaneous detection of contributor and contribution types.

Furthermore, the data collection[21] relied on the well-known MediaWiki,[22] a set of Python utilities for extracting and processing the features in Table 2. The stream-based ML models were incrementally updated and evaluated with `EvaluatePrequential`.[23] For each observed sample, the prequential evaluation makes a prediction, tests and trains the model.

4.1. Feature analysis

To ensure good classification performance, feature pairwise correlation analysis was followed by feature selection.

Data analysis detects the relevant features for contributor profiling. Therefore, the correlation between features and targets has been determined through the Pearson correlation coefficient. Table 3 shows for each target – (1) *user type*, *i.e.*, human or non-human; and (2) *contribution type*, *i.e.*, malevolent or benevolent – the features with a correlation below −0.15 or above 0.15. Note that these results indicate the current problem can be addressed with ML techniques as there exist moderate correlations with both targets.

Feature selection, as explained in Section 3.1.3, applies a combinatorial search over configurable parameter ranges and uses the RFE feature selection algorithm and LinearSVC to determine the best features. At the end, for the first target, the selected features (Table 2) were: #6 to #10, #12, #15 (*good faith*), #16 (E probability) and #18 (B and *stub* probabilities). Furthermore, the second target relies on #18 (excluding GA probability).

4.2. Synthetic data generation

The quality of the new data is established by comparing the statistical properties and class distribution of both original and generated data.

Analysis of the generated synthetic data is shown in Table 4, a statistical comparison between the original and generated data. Specifically, it shows the statistical attributes of the real data and the relative percentage change of the synthetic versus the original samples considering mean, minimum values, the first, second and third quartiles. The results cover all relevant features for both targets, excluding #15 (*good faith*) in Table 2 because synthetic and the original data are statistically identical. Since the variations are minimal for most features, the results indicate that the proposed synthetic data generation algorithm maintains the statistical attributes of the real data. The sole exception is the number of reverted contributions per contributor (#9 in Table 2) since it does not represent a probabilistic value.

Clustering is performed using K-means. Fig. 5 shows the distribution of classes for the experimental data set after adding synthetic information. Note that the user type is determined by the bot flag, feature #2 in Table 2, while the type of contribution is based on the OK probability (feature #17 in Table 2). In the latter case, when the OK probability is above 50 %, the contribution is positive, otherwise it is negative.

[21] Available from the corresponding author on reasonable request.
[22] Available at www.pypi.org/project/mediawiki-utilities, May 2022.
[23] Available at https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.evaluation.EvaluatePrequential.html, May 2022.

Table 4
Statistical comparison between the original and the synthetic data using mean, minimum, the first, second and third quartiles.

Statistics for the original bot user samples												
Feature	#6	#7	#8	#9	#10	#12	#16 (E)	#18 (B)	#18 (C)	#18 (FA)	#18 (Start)	#18 (Stub)
mean	1.55	195.53	137.84	5.03	0.01	0.07	0.94	0.20	0.06	0.01	0.31	0.34
min	1.00	1.00	1.00	0.00	0.00	0.00	0.75	0.01	0.01	0.00	0.02	0.01
Q1	1.03	40.50	30.83	0.00	0.00	0.00	0.94	0.18	0.06	0.01	0.30	0.29
Q2	1.13	77.50	51.75	2.00	0.00	0.00	0.95	0.21	0.06	0.01	0.31	0.32
Q3	1.60	289.75	127.68	6.00	0.01	0.00	0.96	0.22	0.07	0.01	0.33	0.37
% Δ Statistics for the bot user generated samples												
mean	−13.60	−13.36	−41.42	−37.90	1073.63	−100.00	−0.97	25.22	11.86	48.30	−2.74	8.56
min	0.00	0.88	0.66	0.00	0.00	0.00	0.01	0.42	0.14	0.00	0.31	0.91
Q1	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Q2	0.00	0.01	0.01	−25.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Q3	0.00	0.00	0.00	−12.50	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00

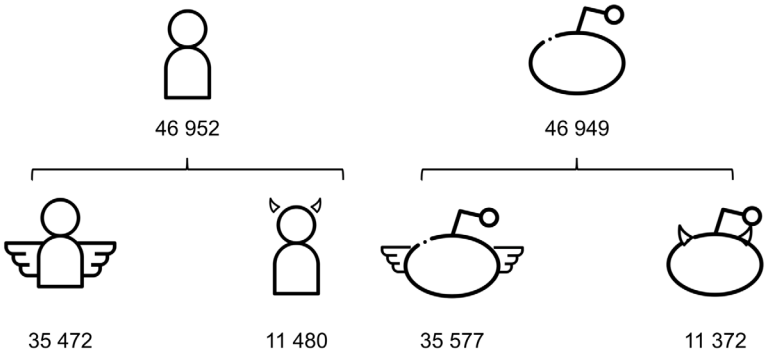

Fig. 5. Distribution of *user type* and *contribution type* targets in the balanced data set.

4.3. Classification of human and bot contributors

The proposed method estimates if the contributor is human or bot using the bot flag target feature. While class #0 represents human contributors, class #1 is bot contributors. This experiment has been performed with a data stream produced from the original class-imbalanced data and three different sets of features:

1. Basic set of features (features #3 to #14 from Table 2);
2. Full set of features (see Table 2);
3. Selected set of features (see Section 4.1).

Table 5 shows the macro and micro results of this classification. The values obtained are consistent in most cases for all classifiers and the values increase when using all features. Furthermore, the proposed method presents promising results when exclusively employing the most relevant features, since it reduces the processing time. The best binary classifier in all experiments is RF. Moreover, even though the original data set is deeply imbalanced, the system achieves a remarkable performance for both human and bot classes.

4.4. Classification of benign and malign contributors

The benign and malign classification experiments comprise: (*i*) stream-based classification of the original class-imbalanced data using binary classifiers; and (*ii*) stream-based classification of the original class-imbalanced and the new class-balanced data using a multi-class RF stacking classifier.

4.4.1. Binary classification

Table 6 provides the macro and micro results for the second target, *i.e.*, benevolent and malevolent contributors classification, with single and ensemble binary classifiers and different feature sets. This experiment was performed with a data stream produced from the original class-imbalanced data set and the three different sets of features.

In light of the results, we can conclude that distinguishing between positive and negative contributions is not as straightforward as identifying humans and bots. Nonetheless, the proposed method achieves 89.32 % accuracy and 84.64 % macro F-measure with the RF classifier for the best feature set analysed.

Table 5
Performance of the human/bot classification.

Set	Classifier	Accuracy	*F*-measure			Time (s)
			Macro	#0	#1	
1	NB	95.55	62.04	97.71	26.37	4.37
	DT	99.52	84.93	99.76	70.09	180.79
	RF	**99.83**	**94.86**	**99.91**	**89.82**	129.78
	BC	**99.83**	94.84	**99.91**	89.77	144.20
2	NB	93.13	57.86	96.41	19.31	7.23
	DT	99.50	83.97	99.74	68.19	387.38
	RF	**99.83**	94.60	**99.91**	89.30	164.06
	BC	**99.83**	**94.67**	**99.91**	**89.44**	263.68
3	NB	97.40	68.55	98.67	38.42	4.21
	DT	99.55	84.91	99.77	70.04	196.16
	RF	**99.82**	**94.40**	**99.91**	**88.89**	138.84
	BC	**99.82**	94.34	99.90	88.77	110.26

Table 6
Performance of the benevolent/malevolent classification.

Set	Classifier	Accuracy	*F*-measure			Time (s)
			Macro	#0	#1	
1	NB	57.50	54.84	65.79	43.90	4.24
	DT	80.10	69.17	87.53	50.82	344.72
	RF	**88.17**	**82.94**	**92.39**	**73.49**	196.26
	BC	78.63	72.50	85.49	59.50	127.39
2	NB	69.74	62.42	79.00	45.85	6.92
	DT	80.66	70.17	87.85	52.50	725.77
	RF	**89.32**	**84.64**	**93.11**	**76.17**	270.83
	BC	80.90	74.59	87.25	61.93	211.41
3	NB	75.03	55.68	84.96	26.41	2.93
	DT	78.95	66.16	86.96	45.35	203.73
	RF	**88.02**	**82.57**	**92.32**	**72.83**	180.11
	BC	79.89	68.85	87.39	50.31	101.09

Table 7
Performance of the multi-class RF stacking classifier versus baseline.

Model	Accuracy	*F*-measure			Time (s)
		Macro	#0	#1	
Baseline[a]	88.02	82.57	92.32	72.83	180.11
Original data set	88.71	83.85	92.71	74.98	977.87
Balanced data set	**91.44**	**87.46**	**94.52**	**80.39**	1834.64

[a]The baseline was extracted from Table 6, set 3.

4.4.2. Multi-class classification

The stacking model allows the simultaneous classification of contributor type (humans versus bots) and contribution type (positive versus negative). This experiment was performed with two data streams generated from the original class-imbalanced and the new class-balanced sets. The goal is to analyse the impact of the balanced data set, comprising original and synthetically generated data.

Table 7 contains the results of: (*i*) the baseline model obtained with feature set 3, the RF classifier and a stream produced from original class-imbalanced data (see Tables 5 and 6); (*ii*) the stacking model with a stream produced from the original class-imbalanced data; and (*iii*) the stacking model with a stream produced from the class-balanced data. When compared with the baseline, the stacking model improves the results in both accuracy and *F*-measure. Furthermore, the results show the positive impact of the class-balanced data set. The difference in processing time is due to the fact that the imbalanced stream has 47,369 events and the balanced stream has 93,901 events. Actually, the process time per event is identical: 21 ms/event for the imbalanced and 20 ms/event for the balanced data stream.

4.5. Literature comparison

The results of the proposed method are herein compared with some of the works listed in Table 1. Most of the related search adopts offline processing. Note that online models are built from scratch and incrementally updated and evaluated, whereas the offline models are trained and then tested using distinct data partitions. Zheng et al. (2019) [22] detect the type of contributor and

contributions with a similar performance in both accuracy and macro F-measure to the proposed method. However, social media data should be treated as streams since new revisions arrive continuously, resulting in instant classifier updates [16,17]. Thus, having comparable results endorses the proposed method.

Taking into account solely contribution type detection, Adler et al. (2011) [5] report 99 % accuracy with a rather low recall metric of 30 %, whereas Choi et al. (2016) [8] match the same accuracy value without additional metrics for fair comparison. Kumar et al. (2016) [14] identify hoaxers from non-hoaxers with 63 % accuracy (26 and 28 percent points lower in accuracy for contribution type detection than our proposal as shown in Tables 6 and 7, respectively). Compared to Green & Spezzano (2017) [15] and Velayutham et al. (2017) [11] contributor type detectors, our solution attains +17 and +9 percent points in accuracy for bot detection (see Tables 5 and 7, respectively). Finally, Zheng, Albano et al. (2019) [19] also report worse results for contributor classification: almost –10 and –3 percent points regarding macro F-measure for bot detection according to Tables 5 and 7, respectively. Since the obtained online results contemplate all data samples rather than just the test partition samples, the proposed stream-based method considerably improves the task of classifying contribution and contributor types. Note that some works were not included in this analysis, *e.g.*, [1,6,7,13,16,17,20], since they do not provide directly comparable metrics.

5. Conclusions

Wiki-based repositories are built by a network of contributors, known as the crowd. The crowd voluntarily shares information related to people, places, entities, *etc*. This voluntary nature of crowdsourcing platforms facilitates unethical behaviours from ill-intentioned human and bot contributors, enabling fraudulent data manipulation to satisfy third party interests. This scenario raises reliability concerns regarding the data quality of wiki-based platforms. To mitigate these issues, this work proposes a stream-based method to automatically classify contributors and their contributions. Specifically, the overall solution is composed of two distinct components: (*i*) offline data pre-processing and data simulation; and (*ii*) stream-based contributor profiling and classification. Pre-processing encompasses data analysis, feature engineering, and feature selection to identify the most promising edition-related features for profiling. The simulation is based on synthetic data generation to solve the deep class imbalance of the original data set. The stream-based profiling incrementally builds the profiles of contributors based on the selected edit features. The classification explores different stream-based binary classifiers with both class-imbalanced and class-balanced data streams to finally choose a stacking multi-class classifier that simultaneously identifies benign and malign humans and bots.

The proposed method was tested and evaluated with a real data set from Wikivoyage holding contributions from 417 bots and 46,952 humans, using classification accuracy and F-measure metrics. The experiments were conducted using the original Wikivoyage data together with synthetic data as streams, which incrementally update contributor profiles and classifier models. The final two-level stacking classifier presents results that vary between 80 % and 95 % concerning accuracy and F-measure, proving the efficiency of the method.

To sum up, this paper describes a solution that can be used to anticipate and isolate malevolent content produced by the identified malign contributors. As future work, the plan is to generate stream-based synthetic data as well as explain classifications, employing Natural Language Processing techniques to generate statements automatically.

Declaration of competing interest

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

Acknowledgements

This work has been supported by: (*i*) Xunta de Galicia, Spain grant ED481B-2021-118, Spain; (*ii*) National Funds through the FCT – Fundação para a Ciência e a Tecnologia, Portugal (Portuguese Foundation for Science and Technology) as part of project UIDB/50014/2020; (*iii*) CHIST-ERA, Ireland and the Irish Research Council, Ireland as part of the "Smart Pharmaceutical Manufacturing (SPuMoNI)" research project [Apr/2019–Dec/2022]; and (*iv*) University of Vigo, Spain/CISUG for open access charge.

References

[1] M. Tsvetkova, R. García-Gavilanes, L. Floridi, T. Yasser, Even good bots fight: The case of wikipedia, PLoS ONE 12 (2) (2017) e0171774, http://dx.doi.org/10.1371/journal.pone.0171774.

[2] R. Egger, I. Gula, D. Walcher (Eds.), Open tourism: Open innovation, crowdsourcing and co-creation challenging the tourism industry, in: Tourism on the Verge, Springer, 2016, p. 476, http://dx.doi.org/10.1007/978-3-642-54089-9.

[3] S. Kumar, J. Cheng, J. Leskovec, Antisocial behavior on the web: Characterization and detection, in: Proceedings of the International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 947–950, http://dx.doi.org/10.1145/3041021.3051106.

[4] T. Steiner, Bots vs. Wikipedians, anons vs. Logged-ins (redux): A global study of edit activity on wikipedia and wikidata, in: Proceedings of the International Symposium on Open Collaboration, ACM, 2014, pp. 1–7, http://dx.doi.org/10.1145/2641580.2641613.

[5] B.T. Adler, L. de Alfaro, S.M. Mola-Velasco, P. Rosso, A.G. West, Wikipedia vandalism detection: Combining natural language, metadata, and reputation features, in: Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing, in: LNCS, vol.6609, Springer, 2011, pp. 277–288, http://dx.doi.org/10.1007/978-3-642-19437-5_23.

[6] G. Yang, S. He, Z. Shi, Leveraging crowdsourcing for efficient malicious users detection in large-scale social networks, IEEE Internet Things J. 4 (2) (2017) 330–339, http://dx.doi.org/10.1109/JIOT.2016.2560518.
[7] V.S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, F. Menczer, The DARPA Twitter bot challenge, Computer 49 (6) (2016) 38–46, http://dx.doi.org/10.1109/MC.2016.183.
[8] H. Choi, K. Lee, S. Webb, Detecting malicious campaigns in crowdsourcing platforms, in: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, IEEE, 2016, pp. 197–202, http://dx.doi.org/10.1109/ASONAM.2016.7752235.
[9] Z. Yamak, J. Saunier, L. Vercouter, Detection of multiple identity manipulation in collaborative projects, in: Proceedings of the International Conference Companion on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 955–960, http://dx.doi.org/10.1145/2872518.2890586.
[10] M. Schonlau, R.Y. Zou, The random forest algorithm for statistical learning, Stata J.: Promot. Commun. Stat. Stata 20 (1) (2020) 3–29, http://dx.doi.org/10.1177/1536867X20909688.
[11] T. Velayutham, P.K. Tiwari, Bot identification: Helping analysts for right data in Twitter, in: Proceedings of the International Conference on Advances in Computing, Communication, & Automation, IEEE, 2017, pp. 1–5, http://dx.doi.org/10.1109/ICACCAF.2017.8344722.
[12] P.G. Efthimion, S. Payne, N. Proferes, Supervised machine learning bot detection techniques to identify social Twitter bots, SMU Data Sci. Rev. 1 (2) (2018) 5:1–20.
[13] A. Rauchfleisch, J. Kaiser, The false positive problem of automatic bot detection in social science research, PLoS ONE 15 (10) (2020) e0241045, http://dx.doi.org/10.1371/journal.pone.0241045.
[14] S. Kumar, R. West, J. Leskovec, Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes, in: Proceedings of the International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 591–602, http://dx.doi.org/10.1145/2872427.2883085.
[15] T. Green, F. Spezzano, Spam users identification in wikipedia via editing behavior, in: Proceedings of the International AAAI Conference on Web and Social Media, AAAI, 2017, pp. 532–535.
[16] S. Heindorf, M. Potthast, B. Stein, G. Engels, Vandalism detection in wikidata, in: Proceedings of the ACM International Conference on Information and Knowledge Management, ACM, 2016, pp. 327–336, http://dx.doi.org/10.1145/2983323.2983740.
[17] S. Heindorf, Y. Scholten, G. Engels, M. Potthast, Debiasing vandalism detection models at wikidata, in: Proceedings of the ACM Web Conference, ACM, 2019, pp. 670–680, http://dx.doi.org/10.1145/3308558.3313507.
[18] A. Sarabadani, A. Halfaker, D. Taraborelli, Building automated vandalism detection tools for wikidata, in: Proceedings of the International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 1647–1654, http://dx.doi.org/10.1145/3041021.3053366.
[19] L.N. Zheng, C.M. Albano, N.M. Vora, F. Mai, J.V. Nickerson, The roles bots play in wikipedia, Proc. ACM Human-Comput. Interaction 3 (CSCW) (2019) 1–20, http://dx.doi.org/10.1145/3359317.
[20] L. Sanchez, E. Rosas, N. Hidalgo, Crowdsourcing under attack: Detecting Malicious behaviors in waze, in: IFIP Advances in Information and Communication Technology, Vol. 528, Springer, 2018, pp. 91–106, http://dx.doi.org/10.1007/978-3-319-95276-5_7.
[21] A. Hall, L. Terveen, A. Halfaker, Bot detection in wikidata using behavioral and other informal cues, Proc. ACM Human-Comput. Interaction 2 (2018) 1–18, http://dx.doi.org/10.1145/3274333.
[22] P. Zheng, S. Yuan, X. Wu, J. Li, A. Lu, One-class adversarial nets for fraud detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI Press, 2019, pp. 1286–1293, http://dx.doi.org/10.1609/aaai.v33i01.33011286.
[23] N. Joshi, F. Spezzano, M. Green, E. Hill, Detecting undisclosed paid editing in wikipedia, in: Proceedings of the ACM Web Conference, ACM, 2020, pp. 2899–2905, http://dx.doi.org/10.1145/3366423.3380055.
[24] X. Zhang, A.A. Ghorbani, An overview of online fake news: Characterization, detection, and discussion, Inf. Process. Manage. 57 (2) (2020) 102025, http://dx.doi.org/10.1016/j.ipm.2019.03.004.
[25] J. Benesty, J. Chen, Y. Huang, I. Cohen, Pearson correlation coefficient, in: Noise Reduction in Speech Processing, in: STSP, vol. 2, Springer, 2009, pp. 37–40, http://dx.doi.org/10.1007/978-3-642-00296-0_5.
[26] Z. Wan, Y. Zhang, H. He, Variational autoencoder based synthetic data generation for imbalanced learning, in: Proceedings of the IEEE Symposium Series on Computational Intelligence, IEEE, 2017, pp. 1–7, http://dx.doi.org/10.1109/SSCI.2017.8285168.
[27] S. Jain, G. Seth, A. Paruthi, U. Soni, G. Kumar, Synthetic data augmentation for surface defect detection and classification using deep learning, J. Intell. Manuf. 33 (4) (2020) 1007–1020, http://dx.doi.org/10.1007/s10845-020-01710-x.
[28] A.R. Kurup, G. Sajeev, A task recommendation scheme for crowdsourcing based on expertise estimation, Electron. Commer. Res. Appl. 41 (2020) 100946, http://dx.doi.org/10.1016/j.elerap.2020.100946.
[29] M. Mukherjee, M. Khushi, SMOTE-ENC: A Novel SMOTE-based method to generate synthetic data for nominal and continuous features, Appl. Syst. Innov. 4 (1) (2021) 18, http://dx.doi.org/10.3390/asi4010018.
[30] F. Salutari, D.D. Hora, G. Dubuc, D. Rossi, Analyzing wikipedia users' perceived quality of experience: A large-scale study, IEEE Trans. Netw. Serv. Manag. 17 (2) (2020) 1082–1095, http://dx.doi.org/10.1109/TNSM.2020.2978685.
[31] G. Amaral, A. Piscopo, L.-A. Kaffee, O. Rodrigues, E. Simperl, Assessing the quality of sources in wikidata across languages: A hybrid approach, J. Data Inf. Quality 13 (4) (2021) 1–35, http://dx.doi.org/10.1145/3484828.
[32] T. Liu, Z. Chen, B. Zhang, W.-Y. Ma, G. Wu, Improving text classification using local latent semantic indexing, in: Proceedings of the IEEE International Conference on Data Mining, IEEE, 2004, pp. 162–169, http://dx.doi.org/10.1109/ICDM.2004.10096.
[33] Y. Liu, H.T. Loh, A. Sun, Imbalanced text classification: A term weighting approach, Expert Syst. Appl. 36 (1) (2009) 690–701, http://dx.doi.org/10.1016/j.eswa.2007.10.042.