Soft Computing (2021) 25:4345–4366
https://doi.org/10.1007/s00500-020-05445-y

METHODOLOGIES AND APPLICATION

An intrinsic authorship verification technique for compromised account detection in social networks

Ravneet Kaur[1] · **Sarbjeet Singh**[1] · **Harish Kumar**[1]

Published online: 1 January 2021

Abstract
The proliferation of social networks resulted in a remarkable increase in their popularity empowering users to create, share and exchange content for interaction and communication among them. However, these have also opened new avenues for malicious and unauthorized use. This paper presents an intrinsic profiling-based technique for the assessment of authorship verification and its application toward detection of compromised accounts. For the same, efficiency of different textual features has been examined. Four categories of features, namely content free, content specific, stylometric and folksonomy, are extracted and evaluated. Experiments are performed with 3057 twitter users taking 4000 latest tweets for each user. Various feature selection techniques are used to rank and select optimal features for each user which are further combined using a popular rank aggregation technique called BORDA. The problem of authorship verification in this paper is studied as a unary classification problem. Performance of various one-class classifiers, namely Local Outlier Factor, Isolation Forest and One-Class SVM, is analyzed on the basis of different evaluation metrics. Experimental results depict that OCC-SVM with rbf kernel outperformed other one class classifiers attaining an average F-score of 87.29% and Matthews Correlation Coefficient of 74.42% under varied parameter settings.

Keywords Authorship verification · Continuous authentication · Compromised account detection · Classification · One class classification · Social networks

1 Introduction

The world presently is experiencing a new era with the presence of social media as one of the greatest communication means. It has blurred the separation between physical and virtual worlds and people nowadays are identified not only by their physical identities but virtual identities as well. Research is going on to propose schemes for efficient and secure handling of this multimedia social data in domains such as recommendation systems (Zhang et al. 2017, 2019b), cyber security (Gupta et al. 2020; Sahoo and Gupta 2019), behavioral profiling (Zhang et al. 2019a; Al-Qurishi et al. 2018), community detection (Al-Ayyoub et al. 2019; Al-Andoli et al. 2020), authorization (Feng et al. 2016; Li et al. 2018), event management (Amato et al. 2019) etc. Though past few years have seen a widespread growth of social media in various domains but it has also opened avenues for a number of sophisticated and malicious attacks such as spread of false and unsolicited information, deceitful advertisements, dissemination of spamming content and many more. Spam accounts responsible for spreading the malicious activity are usually the fake accounts (created solely for this purpose) or compromised accounts (belonging to legitimate account owners but hacked and controlled by some attacker). In order to deal with this issue of malicious activity, researchers have proposed numerous methods to detect the spam and fake accounts (Miller et al. 2014; Chakraborty et al. 2016; Kaur et al. 2018a; Wu et al. 2018; Singh et al. 2018; Inuwa-Dutse et al. 2018; Zhang and Ghorbani 2020; Javed et al. 2019) but only a handful of research is done toward the detection of compromised accounts (Ruan et al. 2016; Egele et al. 2017; Igawa et al. 2016; Kaur et al. 2018b; Velayudhan 2019),

Communicated by V. Loia.

✉ Ravneet Kaur
ravneets48@gmail.com

Sarbjeet Singh
sarbjeet@pu.ac.in

Harish Kumar
harishk@pu.ac.in

[1] University Institute of Engineering and Technology, Panjab University, Chandigarh, India

which is otherwise a topic of huge concern. With compromised accounts, original legitimate users lose access to their accounts and hence detection of such accounts also involves credential recovery as well as return of the authentic access of accounts to the genuine users. Moreover, compromising the legitimate accounts is relatively a safer option for criminals so as to impersonate themselves as the original user thereby hiding their true identity. Techniques such as those proposed by Wang et al. (2019) have been put into practice to evaluate dynamic trust among different nodes but still spammers are getting smart enough to befool the users.

Service providers of popular social networks rely on point-of-entry-based approaches such as IP geolocation-based authentication at the time of login which seems to be insufficient. After the legitimate login into an account, the authenticated session can be easily commandeered by an intruder through various cookie stealing and session hijacking attacks. Hence, the standard static login authentication applied only once and that too at the start of the session should be supplemented with other reliable practices that could be applied unobtrusively at the back end. One such practice is the use of continuous authentication (CA) to re-verify the user during the session. Continuous Authentication of data entails good accuracy with small authentication delay as well as confrontation to manipulation and forgery (Brocardo and Traore 2014; Brocardo et al. 2019). With the presence of CA, even if malicious intruders bypass the initial authentication step, they will be under monitoring every now and then and hence may not be able to freely use the account. This continuous authentication can be applied either at the front end (Active CA) in the form of biometric authentication or unobtrusively at the back end (Passive CA) at stylometric level. It is evident from the literature that use of biometric modalities for continuous authentication has been in wide practice since long but use of stylometry-based CA is still in its embryonic state with only a few works primarily dedicated toward it (Zheng et al. 2006; Brocardo et al. 2015; Kocher and Savoy 2017; Brocardo et al. 2017, 2019). Though active continuous authentication mechanisms provide better accuracy but it often infuriates the users because of them being repeatedly involved in the process. On the other hand, passive authentication mechanisms do not suffer from this drawback and hence can be deployed inconspicuously at the back end. In this work, stylometry-based continuous authentication has been studied and analyzed.

From academic research point of view, Behavioral Profiling has been the key procedure adopted for the detection of compromised accounts where continuous authentication process involves the verification of similarity between the current behavior and the behavioral profile pattern already stored in for a user. While behavioral profile could be built considering different modalities such as text, meta-data, temporal patterns, introversial and extroversial usage patterns etc., the major objective of this paper is to perform text-based continuous authentication and investigate whether profiling user's textual and stylometric behavior could help detect the authenticity of an incoming message from a user profile. Furthermore, this paper focuses on the viability of continuous authentication process for the detection of compromised accounts and accordingly the problem has been structured as a document representation model. This paper examines the task of continuous authentication on short texts and applies text mining for stylometric analysis of short excerpts of social media text. The same has been performed in this work using authorship verification process which involves assessing whether the text in question T_x is written by the concerned user (X) or not when only a subset of reference texts R_x of the user X are given.

Most of the existing works in the literature have studied the problem of authorship verification aka continuous authentication as a binary classification problem taking data samples of same user as the positive class and data samples belonging to spam messages or other random users as negative class (Trång et al. 2015; Seyler 2018; Kaur et al. 2018b). Though this has been a widely adopted policy to create a synthesized negative class data, yet it is often unlikely to obtain a very-well sampled negative data. Obtaining the negative class in case of problems such as compromised account detection is limitless and it is often difficult to cover all the possible abnormal behaviors. Also, with this limited negative class data, any new abnormal behavior not previously learned in the training phase would not be correctly detected at run time. Hence, in this paper, the problem has been studied from one-class classification perspective where only the data samples from the same user have been used for profiling and training of models. Also, efficiency of three generic commonly deployed one-class classifiers namely, Local Outlier Factor (Breunig et al. 2000), Isolation Forest (Liu et al. 2008) and One-Class SVM (Schölkopf et al. 2000), has been analyzed in order to yield the best classifier to be considered for the automated process.

Following research aspects are covered in this paper:

(1) Analysis of the efficiency of various textual features for continuous authentication and identification of compromised accounts in social networks.
(2) Selecting optimal features for each individual user for further classification.
(3) Examining the viability of one class classification techniques for authorship verification of social network content.

The rest of the paper is structured as follows. Section 2 discusses the work relevant to the authorship verification of online content with the focus on works that made use of text-based features. Section 3 covers the proposed framework,

features and one-class classification approaches adopted for the verification and detection of compromised accounts followed by the demonstration of experimental results in Sect. 4. Finally, Sect. 5 concludes the paper discussing some future insights into the work.

2 Background study and related work

Some of the existing techniques for authorship verification and continuous authentication have encouraged the use of statistical approaches, thereby making a decision based on a user-defined threshold value (Koppel and Winter 2014; Green and Sheppard 2013; Seidman 2013; Halvani and Steinebach 2014; Neal et al. 2018). This process limits the model for flexibility and dynamic updates. Recent works have also seen the engagement of machine learning models to automate the process of authorship verification. Authorship verification actually represents a (unary) one class classification problem (Halvani et al. 2018b) but still in the existing literature the problem has been misapprehended as a (binary) classification problem (Barbon et al. 2017; Brocardo et al. 2019). In AV problems, there is only one class (that of the respective user) to learn from. But in order to train a supervised classification model for a two-class classification problem, there is a need to gather data and extract features from respective labeled classes, i.e., user and not user. Research works that study AV as a binary classification problem consider data from other users to artificially treat such samples as the negative class data. Numerous works in literature have been seen adopting this practice of creating ground truth data to validate the model and analyze the efficiency of various supervised machine learning classifiers (Trång et al. 2015; Seyler 2018; Kaur et al. 2018b; Van Der Walt and Eloff 2018).

2.1 Authorship verification: binary classification perspective

Brocardo et al. (2015) performed continuous authentication on e-mail and Twitter data using lexical, syntactic and application centred textual features. This work superseded their previous work (Brocardo and Traore 2014) reducing the Equal Error Rate [a point with same value for False Acceptance Rate (FAR) and False Rejection Rate (FRR)] by using Information Gain and mutual information for feature selection. Apart from Logistic Regression (LR) and SVM, a hybrid classifier was built by integrating the efficiency of LR and SVM. Experiments were performed by varying block sizes and number of blocks for each user. As a result, different error rates were observed. Overall, highest accuracy was attained by LR classifier followed by Hybrid-SVM and SVM, respectively.

Later, Brocardo et al. (2017) themselves examined the glitch of machine learning architectures for AV and analyzed how deep belief networks attained more promising results (an error rate of 5–12%). Gaussian Bernoulli Deep-Belief Network (GB-DBN) has been used by researchers with a layered framework of Restricted Boltzmann Machines (RBMs) followed by a supervised machine learning classifier. Again as an extension to this work, in Brocardo et al. (2019) experiments were performed with various shallow and deep learning classifiers to investigate the use of various stylometric features.

Likewise, Igawa et al. (2016) and Barbon et al. (2017) proposed an approach for AV utilizing only char *n*-gram features. Using interspersing three sets, namely baseline, threshold and test set, were built. Simplified Profile Intersection (SPI) was used to verify the authenticity of tweets on the social networking platform, Twitter. SPI values of positive and negative test sets were evaluated using an instance-based classifier (kNN). Also, a FIFO policy was adopted to dynamically update the baseline profile which helped to improve accuracy and reduce outliers.

Further Li et al. (2016) examined the efficiency of different stylometric and social network features for AV of short length messages on Facebook. A comparative analysis of different machine learning classifiers was done and a voting algorithm was used to combine the output of various classifiers. As social network messages are shorter in length, researchers mentioned that char-based *n*-grams helped to build a better profile of users than word and sentence-based features attaining an accuracy of 76%. Only 30 users were taken for experiments which is a very poor sample in comparison with the millions of active users worldwide.

Instead of the common consideration of *n*-grams as byte-based, Peng et al. (2016) used bit-level *n*-grams for authorship attribution and verification of online social media data. Moreover, unlike other research works that removed most frequent or infrequent *n*-grams before analysis, here researchers considered all the *n*-grams. Moreover, topic independent data along with informal language such as emoticons and abbreviations were analyzed. Attribution was handled as a classification problem whereas verification as an outlier detection problem with Interquartile Range (IQR).

Kaur et al. (2018b) analyzed the competence of various textual features for detecting compromised accounts in online social networks. Alongside various traditional textual features some stylometric, folksonomic and topic modeling features were used. It was inferred that users did not stay consistent on same set of features; hence, AHP-TOPSIS was used to rank features for each respective user. Accordingly, comparison was made and decision was taken as per the best feature. The problem was studied as a classification problem deploying kNN classifier for the task. Though, an in depth analysis for each user was performed but there remains a

need to carry out the experiments with some more classification techniques, additional users as well as more train and test samples.

Overall from the binary classification perspective, most of the researchers have handled the authorship verification problem using supervised machine learning algorithms, namely kNN, Naive Bayes, Support Vector Machines, Decision Trees, Neural Networks and different deep learning architectures (Brocardo and Traore 2014; Brocardo et al. 2015, 2017, 2019; Barbon et al. 2017; Li et al. 2016; Peng et al. 2016; Kaur et al. 2018b). Also, most commonly deployed features include *n*-grams and stylometric features.

2.2 Authorship verification: unary classification perspective

Techniques discussed in Sect.2.1 have considered authorship verification as a two-class classification problem considering samples from other users as negative class data thereby creating an artificial negative class. Only few of the recently developed techniques have studied the problem in its actual unary form considering it to be a one class classification problem.

Halvani et al. (2018b) identified three type of models, namely unary, binary-intrinsic and binary-extrinsic. For the stylometric AV task, authors analyzed the efficiency of eleven unary and binary verification techniques with specific reference to four generic unary classification algorithms, namely, One class Nearest Neighbor (OCNN), One class Support Vector Machine (OSVM), Local Outlier Factor (LOF) and Isolation Forest. For a corpus of text documents collected from Reddit and Amazon Reviews, it was analyzed that distance-based unary classification algorithms outperformed other methods and interestingly LOF was able to achieve efficiency higher than other methods such as OSVM. Halvani et al. (2018b) also classified AV methods on the basis of determinism and optimizability parameters.

Further, Halvani et al. (2018a) also developed an One class Nearest Neighbor (OCNN) inspired OCCAV method. Instead of feature vectors, text documents were indeed represented as byte streams compressed using PPMd (Prediction by Partial Matching) technique. Also, compression-based cosine (CBC) measure was used to compute the dissimilarities between documents. The document in question is compared in terms of cosine dissimilarity with each existing document of the user and the one with least dissimilarity (d_{min}) is taken as the nearest neighbor document (D_{near}). Again the average value of dissimilarity between D_{near} and each of the remaining documents is measured. If d_{min} comes out to be less than d_{avg}, the unknown document is considered to be written by the concerned author.

Neal et al. (2018) also handled continuous authentication of blog data as a one-class classification problem deploying Isolation Forest classifier for the task. Unlike other related tasks, this work emphasized on the use of small sample and training size data. Character as well as lexical level features were used and Chi-square test was used to study the dependencies between features and the respective class. Linguistic characteristics were seen to be consistent over time and even in the worst-case scenario, a true positive rate of 93% was achieved.

Koppel and Schler (2004) proposed an approach called Unmasking which according to them represented a unary method (Halvani et al. 2018b). But in actual it utilizes the efficiency achieved by binary SVM classifier to remove the irrelevant features and thereby classify the curves into one of the two classes (same-author or different-author). The technique thus cannot be exactly stated to fall under the category of unary methods.

Jankowska et al. (2013) performed authorship verification using proximity-based one class classification approach. Common *n*-gram (CNG) dissimilarity was used to compare the difference in frequencies of *n*-grams. Average dissimilarity ratio was used as an overall measure to classify the document as belonging to the claimed user or not using a similarity score ranging between 0 and 1.

While studying the problem as unary (one-class) classification problem, researchers focused on the deployment of various off-the-shelf one-class classifiers for the task. With the unary perspective, the need for ground truth data is diminished and a better trained model is built which is irrespective of the limited negative class data being learned as in case of binary classification. In this paper also, unary classification has been used for continuous authentication and thereby identification of compromised accounts taking into account the most commonly deployed textual features supplemented with some topic modeling and folksonomic features not used in the literature for the said task.

3 Proposed approach

This section discusses the proposed methodology adopted for the enhanced continuous authentication of social network messages. A one class classification learning approach has been adopted which involves the common steps beginning from data collection, feature extraction and selection to classifier training and model evaluation.

3.1 Approach overview

This research aims to investigate and analyze the efficiency of textual features for continuous authentication (CA) of social network messages. CA problems are handled in two forms: either using a profile-based or an instance-based approach (Stamatatos 2009). In the profile-based approach, all the text

samples of a user are concatenated to form a behavioral profile. Text-based features are extracted from this profile and the unknown text sample is then compared to the profile features to determine its authorship. On the other hand, instance-based approaches take each sample of text separately. Unknown text sample is compared to each text instance separately and answers are usually combined to give a final outcome.

Continuous authentication approach used in this paper is a hybrid approach taking concepts from both profile and instance-based methods. The hybrid approach followed in the work has been illustrated in Fig. 1. Tweets from the popular micro-blogging portal Twitter are used as text samples and the problem is studied as a one-class classification problem because of the presence of tweets from a single class only. Ground truth data in case of problems such as compromised account detection is difficult to obtain because the presence of negative class samples (behavior post compromisation is difficult to acquire as the behavior is either removed once a user gets to know that his account has been compromised or the behavior is not released in public). Hence, in practical scenarios, it is often unlikely to obtain a very well-sampled negative class. Moreover, it is quite difficult and impractical to intentionally compromise or let an account get compromised to obtain compromised data. Secondly with supervised learning, the negative class in case of problems such as compromised account detection, fraud detection etc. is limitless. We cannot cover all the possible abnormal behaviors as there will always remain some negative examples not accounted for. Also, with this limited negative class data, any new abnormal behavior not previously learned in the training phase would not be correctly detected at run time. Because of such limitations, one-class classification has been used in this work to train models from only the positive class data. For a new unknown tweet sample, the objective is to determine whether the activity under consideration has been performed by the concerned user X_i or not. Proposed approach (as shown in Fig. 1) is divided into following steps:

1. *Interspersing*For each user, a set of tweets is collected and placed in two sets namely, Baseline Set ($S1'$) and Training Set ($S2'$) in an interspersed manner. Interspersing has been preferred over direct partitioning in order to place related tweets concerning the similar subjects and trends in both the sets which help overcome the problem of seasonality. Baseline set ($S1'$) acts as the behavioral profile of the user. All the further incoming tweets are matched against this behavioral profile to extract features and similarity measures. On the other hand, features extracted from the training set ($S2'$) help train the machine learning classifiers.
2. *Textual feature extraction* After pre-processing, textual features used in this work (Table 1 are extracted and profiled from all the tweet samples in ($S2'$) of a user. Each user must have enough samples so as to have sufficient amount of labeled training data.
3. *Numerical feature extraction*For experiments, each tweet in the set $S2$ (Training Set) is compared against this baseline profile ($S1$) to compute the numerical features. Textual features extracted from the positive training data samples from ($S2'$) are compared against the Baseline Profile ($S1'$) data to obtain similarity score values calculated using similarity metrics. These values further act as numerical features which help train respective classifier.
4. *Feature selection and rank aggregation* Similarity features extracted are then fed to various feature selection algorithms to score and rank the features. Computed scores and ranks are utilized by the adopted rank aggregation procedure (BORDA) to generate a final ranking of features. Topmost $r\%$ features are selected for further computation.
5. *Supervised machine learning classification* Efficiency of various one class classifiers is analyzed by training the classifiers on performing the k-fold cross-validation on the collected data set. Parameters for each classifier are tuned using a train and validation set and the best fine-tuned classifier is deployed on test set to check the performance.
6. *Testing of new tweet samples* Similar features extracted from the unknown sample are compared with positive (User X_i) learned patterns in the behavioral profile. Accordingly, with the help of suitable machine learning classifier, the unknown tweet sample is assigned the respective class label. If the feature value does not match the positive class, it is counted as an anomalous behavior and the account is detected to be probably compromised. Unlike other domains, here the classifiers are trained and tested independently for each respective user because of the consistency maintained by a user on his/her own behavioral profile.
7. *Continuous Updating of Profiled data*For incorporating the dynamics and changing behavior of users, an incremental learning process is adopted by continuously updating the training data with new patterns. Whenever a new upcoming tweet is predicted by the model to be from the concerned user, continuous updation in the baseline profile and training set will be performed in an interspersed manner. A FIFO approach needs to be followed in which once a new tweet is entered, the oldest tweet present in the original set is discarded in its place. Once, the efficiency of various models is checked on the train-test sets and an efficient model is selected, the updation policy is adopted which is then furthermore used throughout the deployment of the model. This updation of profile and usage of incremental learning is essential to be adopted to keep the process synchronized toward the user's writing style. Once people tend to change the subjects, and maybe

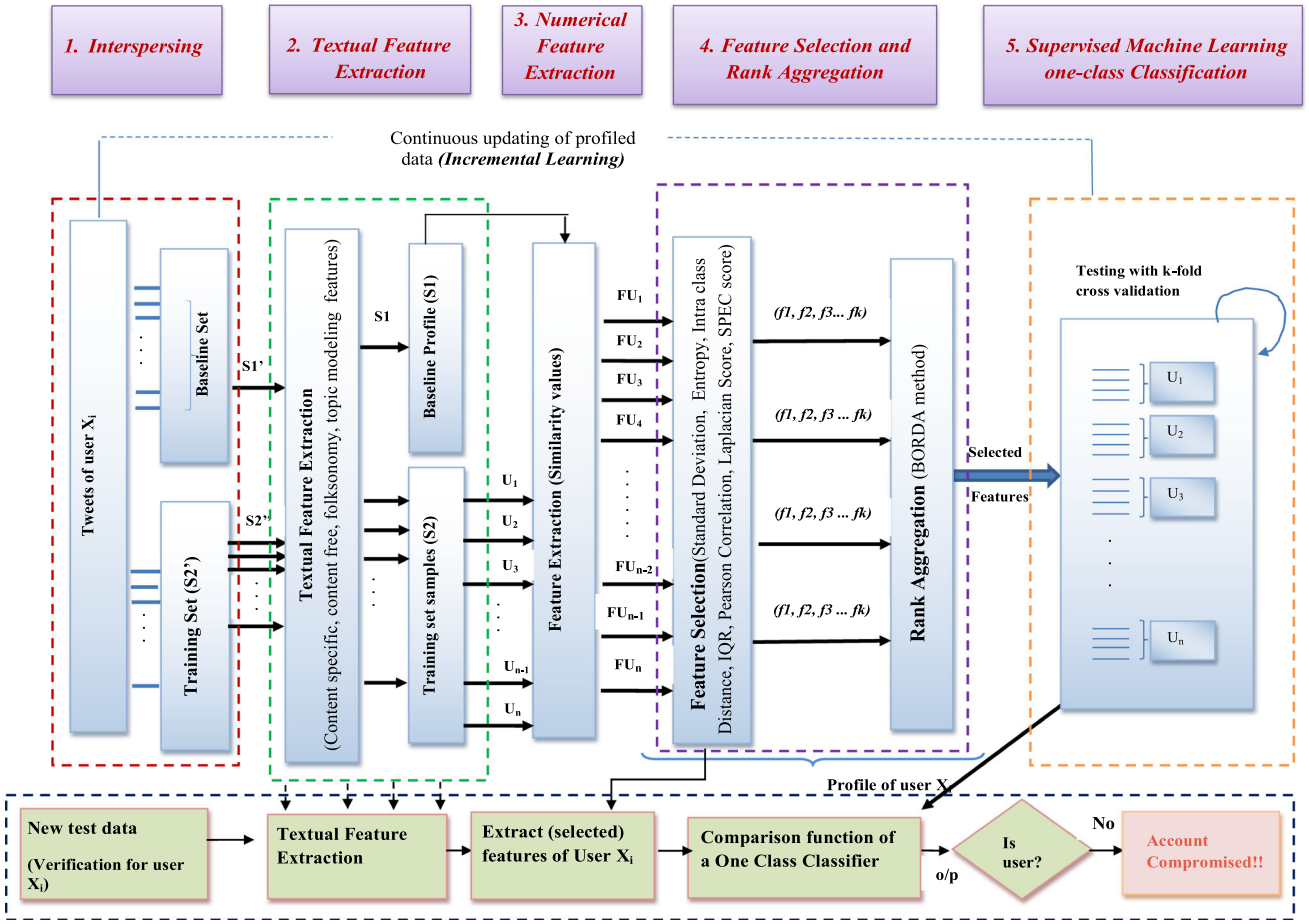


Fig. 1 Framework for intrinsic profiling-based technique for authorship verification and compromised account detection

even writing style over time, it is necessary to use the most recent posts from a user to recognize him. Thus, the model always discards the oldest posts used in exchange for the recently recognized ones.

3.2 Feature engineering

This work involves the use of both content-free and content-specific textual features for profiling and representing the writing style of a user. For the former, *n*-gram, bag of words (BOW) and folksonomic representation have been studied whereas in the latter case stylometric features have been examined.

3.2.1 Feature extraction

Both content-free as well as content-specific features are extracted for a tweet sample and represented as a feature vector. Features have been normalized using RobustScaler[1] in order to make a balance between values of different features. RobustScaler normalizes the features in the range of 0 to 1 by replacing the feature value using $\text{val}_{\text{new}} = (\text{val} - Q_1(\text{val}))/(Q_3(\text{val}) - Q_1(\text{val}))$ where Q defines the interquartile range. This normalization is preferred to reduce the effect of outliers which may be present in the data. A total of 124 features have been extracted which are placed under 28 feature categories as shown in Table 1.

Content-specific features

These features analyze the actual content of the message and examine the meaning of text. Features explored in this category are *n*-grams (char and word *n*-grams), bag of words (BOW) and topic modeling features. In *n*-gram analysis, both char and word *n*-grams have been extracted.

n-gram features process '*n*' consecutive units of text together. Units could be anything ranging from characters and words to bits and bytes. In this paper, both character as well as word-based *n*-grams have been studied. As evident from prior literature (Barbon et al. 2017; Kaur et al. 2018b), char *n*-grams with $n = 6$ has shown good performance. Hence, in this work also, experiments have been performed

[1] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html.

Table 1 Textual features and the corresponding similarity metric

Baseline feature	Notation	Similarity feature metric
Char *n*-grams	R0	Simplified profile intersection (SPI)
	R1	Jaccard coefficient (JC)
Unigram	R2	Simplified profile intersection (SPI)
	R3	Jaccard coefficient (JC)
Bigram	R4	Simplified profile intersection (SPI)
	R5	Jaccard coefficient (JC)
Token prefix	R6	Simplified profile intersection (SPI)
Token suffix	R7	Simplified profile intersection (SPI)
Token *n*-gram prefix	R8	Simplified profile intersection (SPI)
Token *n*-gram suffix	R9	Simplified (SPI)
BOW model	R10	SPATIUM L1
	R11	tf-itf
Topic modeling/Folksonomy	R12	Non-negative matrix factorization (NMF)
	R13	tf-itf
Stylometric		
Alphabetic characters	R14	Cosine similarity
Numeric	R15	
White spaces	R16	
Special characters	R17	
Punctuation	R18	
Frequency of alphabetic, numeric, whitespace,	R19	
Special and punctuation characters		
Social network characters	R20	
Placement of social network characters	R21	
Consecutive characters	R22	
Length of words of different characters	R23	
HapaxLegomena and DishapaxLegomena	R24	
Function words	R25	
Dominant words (Frequent words used by the user)	R26	
All features (Combined)	R27	

with $n = 6$. On the other hand, for word *n*-grams, both unigrams ($n = 1$) and bigrams ($n = 2$) have been examined.

With the **Bag of words** feature model, a data-driven method is used in which frequency count of the most frequently occurring words is analyzed using two type of frequency features namely, term-frequency inverse tweet frequency (tf-itf) and SPATIUM-L1 (Kocher et al. 2016). tf-itf measures the count of occurrence of a word in a tweet alongside the count of tweets in which the word occurs. Similarly, a distance measure called SPATIUM-L1 as defined in Equation 1 computes the distance value using occurrence probability of the frequent term in the given tweets using manhattan distance.

$$\mathrm{SL}(U, V) = \sum_{j=1}^{h} \left(P_U[w_j] - P_V[w_j] \right) \tag{1}$$

where 'U' and 'V' denote the tweet in question and baseline set, respectively. Also, w_j defines each frequently occurring term amongst a total of 'h' frequent terms with P_U and P_V representing the probabilities.

Topic modeling features are used to profile the information related to the popular topics tweeted by a user. This has been performed using folksonomic (hashtags) and Non-Negative Matrix Factorization (NMF) concepts. Folksonomy also referred as social tagging uses two popular Twitter entities namely hashtags and mentions. Frequency counts of popular hashtags/mentions and NMF extracted topics are profiled and compared against the incoming tweets using similar tf-itf concepts as in Bag of words model.

Content-free features

Second set of text-based features explored in this work are **stylometric features**. Content-specific features focus on the

meaning of the text whereas content free features on the other hand deal with how the content is written, i.e., analyze the writing style patterns of users. Both lexical and syntactic stylometric features are used for the task. Lexical features analyze the frequency of occurrence of various words, characters, punctuations, white spaces, legomenas and other vocabulary richness measures whereas syntactic features include the punctuations, function words, parts of speech etc.

Similarity/comparison measures

In this work, machine learning classifiers are fed with the numerical values of attributes which are obtained by applying the appropriate similarity measures on the textual features. Table 1 presents the feature and the corresponding similarity metric used. Similarity is computed taking each tweet in training and testing set as data tweet and all the tweets in baseline set as reference set.

Simplified profile intersection (SPI): SPI is a commonly deployed intersection measure used to define the count of common units between any compared sets and is calculated as follows:

$$\mathrm{SPI}(U, V) = |N(U) \cap N(V)| \tag{2}$$

where U and V are two sets with some textual elements. SPI can attain a minimum value of 0 indicating that there is nothing common in between the sets whereas the maximum value equivalent to the size of the smallest set out of U and V indicating that the bigger set contains whole of the smaller set.

Jaccard coefficient (JC): It is also a similarity measure defined over the intersection (common) and union (all) of elements present in the sets. Again for two textual sets U and V, Jaccard coefficient is calculated as:

$$\mathrm{JC}(U, V) = |N(U) \cap N(V)|/|N(U) \cup N(V)| \tag{3}$$

As Jaccard coefficient is directly proportional to the count of common elements, more the shared elements, higher is the Jaccard coefficient. A user typically adhere to his/her profile, hence, for the authorship verification task, a text from the same user is expected to have higher JC than from other users.

tf-itf As already defined, tf-itf measures the count of occurrence of a word in a tweet alongside the count of tweets in which the word occurs. It is an important and useful measure especially in authorship verification scenario as it keeps into account both the tweet in question and other extracted tweets of the user.

Cosine similarity As stylometric features deal with frequency counts, hence, the cosine similarity method compute similarity by considering features as vectors taking not only frequency count but also similarity of words into account.

SPATIUM-L1 Unlike other discussed measures, SPATIUM-L1 is a distance measure that computes the distance value using occurrence probability of the frequent term in the given tweets (Kocher et al. 2016). Below mentioned is the formula used to calculate SPATIUM-L1

$$\mathrm{SL}(U, V) = \sum_{i=1}^{t} (P_U[w_i] - P_V[w_i]) \tag{4}$$

where t denotes the count of frequently occurring terms taken together. $P_U(w_i)$ and $P_V(w_i)$ define the probability of occurrence of w_i in U and V, respectively. As it is a distance measure, hence a lower value is expected and preferred for the same user and higher for the other users.

3.2.2 Feature selection

Feature selection acts as a crucial and significant step in machine learning process be it unary, binary or a multi-class classification problem. Usually, not all the features extracted for the classification process are important, hence, it becomes necessary to rule out the insignificant and redundant features which otherwise if present may deteriorate the performance of a classifier. Thus, feature selection helps to find those features which represents the most important aspect of data thereby reducing the dimensionality. Three key aspects of a feature selection process are to:

(i) Simplify the classifier by selecting only the relevant features.
(ii) Either improve or at least not reduce the classifier performance.
(iii) Reduce data dimensionality thereby cutting down the dimensions.

In this paper, seven feature selection techniques some of which have been encouraged by Lorena et al. (2015) and seem suitable for one class classification have been used.

Usually in a classification problem, feature importance is measured incorporating the class label into account. But for One Class problems, same label is assigned to all the training examples, hence, such feature selection techniques that are independent of the class label are used. Secondly, with few adjustments, feature selection techniques suitable for unsupervised clustering problems on unlabeled datasets could be deployed for OCC problems. Adjustments in the way that unlike unsupervised learning where multiple clusters need to be distinguishly built from the data, for OCC problems, features able to enhance the distinguishness of only one class need to identified. Each of the adopted feature selection measures in this work produces a ranking of features with importance defined from different views. Hence in order to incorporate data from multiple perspectives, ranking

produced by different feature selection measures are combined. Instead of dealing with the best features obtained by each feature selection technique, rank aggregation using BORDA (de Borda 1784; Lorena et al. 2015) has been performed for aggregating the ranks obtained by the following feature selection techniques.

Standard deviation Standard deviation is a probability-based statistical method calculating the dispersion or variation of features from the mean value. High values of standard deviation signify that values are dispersed over quite a large range whereas a lower value signifies concentrated values close to the mean. It selects features on the basis of feature dispersion. In a one class classification problem, lower the dispersion of values, i.e., standard deviation better is the feature. Mathematically, Standard deviation of a feature is calculated as follows:

$$\mathrm{SD}_f = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_{if} - \bar{x_f}^2)} \tag{5}$$

where x_{if} represents the value of ith data point of fth feature.

Entropy Entropy defined as the "statistical measure of uncertainty" helps to determine the intra-class distribution. In view of the feature selection process, entropy can be used to select such features which minimizes the uncertainty, i.e., entropy thereby imparting the clustering or unsupervised properties making it suitable especially for a one-class classification process.

For $x_1, x_2, x_3 \ldots x_m$ data points defined over a large set Z of data points, the probability distribution φ assigns a non-negative probability $\varphi(x)$ to each data point summation of which amounts to one. For each feature, average values termed as Expectation values in Entropy terms is calculated as follows:

$$\varphi(f_i) = \sum_{x \in Z} \varphi(x) f_i(x) \tag{6}$$

The empirical distribution of a set of sample data points $x_1, x_2, x_3 \ldots x_p$ drawn from Z is defined as:

$$\widetilde{\varphi}(x) = \frac{|\{1 \le j \le p : x_j = x\}|}{p} \tag{7}$$

Accordingly, empirical average of each feature f_i under $\widetilde{\varphi}$ is represented and calculated as:

$$\widetilde{\varphi}(f_i) = \frac{1}{p}\sum_{j=1}^{p} f_i(x_j) \tag{8}$$

$\widetilde{\varphi}(f_i)$ is taken as an estimate of $\varphi(f_i)$ as expectation of a feature f_i under $\widetilde{\varphi}$ and is equivalent to its respective empirical average. Finally, the entropy of $\widetilde{\varphi}$ is calculated as:

$$\mathrm{Entropy}(\widetilde{\varphi}) = -\sum_{x \in Z} \widetilde{\varphi}(x) ln \widetilde{\varphi}(x) \tag{9}$$

For an OCC problem, entropy only output lower values when data points are similar and high values in other cases, i.e., entropy is inversely proportional to the similarity value as higher similarity indicate a structured dataset.

Intraclass distance This measure calculates the average of manhattan distance from all the data points to a centroid value (median value in our case) as defined in Eq. (10).

$$\mathrm{ICD} = \frac{1}{n}\sum_{i=1}^{n}(v_i - v_{\mathrm{med}}) \tag{10}$$

Median is chosen as the centroid value over mean as median is more robust to outliers and it is highly desired that the presence of some anomalous value do not affect the distance measure. The distance measure used in this work is the Manhattan distance which is the difference in value of two data points.

In one class classification problem, positive data points are desired to be closer to one another, hence, lower values of intra-class distances are preferred as such values approximate the data more.

Interquartile range (IQR) In IQR, distribution of values of each feature is studied using the concept of interquartiles. For a feature to effectively represent a class, its values should be highly concentrated which is best reflected using interquartile ranges. For visualization, box plots are often used to represent the interquartile distribution. Though interquartiles of two features may be same when they have overlapping values but overlapping is only present in case of widely dispersed values.

Pearson correlation coefficient Lorena et al. (2015) used Pearson-correlation-based feature selection technique for one-class classification. Pearson measure helps to detect the linear relation in features by calculating the pair wise distance among them. Absolute values of features are taken in order to consider correlation in both direct as well as inverse direction. As shown in Eqs. (11) and (12), for each feature, Pearson correlation coefficient with all the other features is calculated and the absolute values are aggregated. Each feature-feature correlation values of Pearson correlation coefficient range between -1 to 1 but aggregated values may exceed the range as it reflects the overall behavior of each feature. Features having low PC values are considered better and selected during the feature selection process as high values indicate highly correlated feature and inclusion of all such features is

not desired.

$$\mathrm{PC} = \sum_{j=1}^{n} |p_{\mathrm{corr}}(f_i, f_j)| \tag{11}$$

where p_{corr} is defined as follows:

$$p_{\mathrm{corr}}(a, b) = \frac{\sum_{i=1}^{m}[(a_i - \bar{a})(b_i - \bar{b})]}{\sqrt{\sum_{i=1}^{m}(a_i - \bar{a})^2 \sum_{i}^{m}(b_i - \bar{b})^2}} \tag{12}$$

where $\bar{a}$ and $\bar{b}$ represent the mean value of data points in feature a and b, respectively.

Laplacian score Laplacian works on the principle that data points belonging to the same class are usually closer to one another. Accordingly, significance of each feature is assessed using its locality preserving power. Unlike standard deviation and variance which acts as useful measures to represent data, laplacian score helps to distinguish features based on their local structure. It accentuate on the concept that if data points are close to each other, they probably belong to the same topic. Computing the laplacian score of a feature involves the following steps:

(i) For two data points, x_i and x_j, the local structure of data is represented by an affinity matrix S_{ij} which is created by assigning $S_{ij} = \mathrm{e}^{-\frac{|x_i - x_j|^2}{t}}$ if x_i is one of the nearest neighbors of x_j and $S_{ij} = 0$ otherwise. S_{ij} is often referred to as an affinity matrix and represents the local structure of data.

(ii) Laplacian Score is calculated keeping in view the minimization of following objective function

$$\mathrm{Lap}_{\mathrm{score}} = \frac{\sum_{ij}(f_{ti} - f_{tj})^2 S_{ij}}{\mathrm{sd}_{\mathrm{var}}(f_t)} \tag{13}$$

where $\mathrm{sd}_{\mathrm{var}}(f_t)$ defines the standard variance of the t^{th} feature.

SPEC score In spectral analysis also, an affinity matrix (S) representing the similarity values is created for all the data points. To calculate similarity in spectral score, a radial basis function is deployed to the graph structure as defined in Equation (14).

$$S_{ij} = \mathrm{e}^{-\frac{|x_i - x_j|^2}{2\sigma^2}} \tag{14}$$

Similar objective function as that of Laplacian score is used which is as follows:

$$\mathrm{SPEC}_{\mathrm{score}} = \frac{\sum_{ij}(f_{ti} - f_{tj})^2 \mathrm{e}^{-\frac{|x_i - x_j|^2}{2\sigma^2}}}{\mathrm{sd}_{\mathrm{var}}(f_t)} \tag{15}$$

In this measure also, data points closer to each other often have similar values and features can thus be ranked as per relevance, i.e., consistency and similarity of the data points.

3.2.3 Feature fusion/rank aggregation

Rank Aggregation also termed as feature ensemble approach involves the fusion of ranks generated by different feature selection techniques (Namsrai et al. 2013; Shen et al. 2012). Technically, Rank Aggregation can be defined as

*For a given set of objects $o_1, o_2, \ldots, o_n$ and their corresponding rankings $r_1, r_2, \ldots, r_n$, a single ranking **r** is to be produced that is in agreement with the existing rankings*

In this work, objects correspond to the different feature selection algorithms and the rankings to the respective ranks given to the features by each method. As each feature selection technique incorporates different criteria for selecting features, hence aggregating ranks produced by different techniques enables us to analyze various aspects of data considering numerous views of the importance of features. Moreover, ensemble of rankings from multiple techniques help to study the complementaries of various methods and improve the selection by limiting the influence of every single technique.

Literature reveals the existence of numerous methods to combine the rankings produced by different techniques such as aggregation (Wald et al. 2012), majority voting (Tsymbal et al. 2005), Borda (de Borda 1784; Lorena et al. 2015), Schulze (Prati 2012) and many others (Dwork et al. 2001; Tsymbal et al. 2003). In the literature, no consensus is found about which method to prefer over the other. Still taking evidence from some of the research works (Serrai et al. 2017; Sageder et al. 2019; Saari 2001; Singh and Sharan 2015; Lorena et al. 2015), this study involves the use of a popular rank aggregation method named BORDA (de Borda 1784; Lorena et al. 2015) which has been commonly deployed for the rank aggregation task in many interdisciplinary domains. Moreover, a comparative analysis of different rank aggregation methods has been performed by various researchers (Serrai et al. 2017; Singh and Sharan 2015) that suggested the competence of BORDA over other aggregation methods.

BORDA method BORDA (de Borda 1784; Lorena et al. 2015) utilizes a positional voting system by assigning points to each feature according to its ranking position. Points are placed in an increasing order with first ranked feature given lower points and the feature ranked last attaining the maximum points. During decision process, the feature having lowest aggregated points is ranked and hence considered better. The procedure adopted for the same has been illustrated in Algorithm 1. Rather than considering the best feature from this rank aggregation technique as a standalone feature,

experiments have been performed by varying the selected features.

Algorithm 1 BORDA Algorithm for Rank Aggregation

Input: Rankings produced by respective feature selection algorithm (r1, r2, ... r_m)
Output: *Ranks* array corresponding to the ranking produced by BORDA aggregation
Initialization :
/ n is the number of features to be ranked /**
1: **for** $i = 1$ to n **do**
2: feature[i] = 0
3: **end for**
Assignment of positional weights to each feature
4: **for** $i = i$ to n **do**
5: feature[i] = $pos(r_1(i)) + pos(r_2(i)) + pos(r_3(i)) + ... + pos(r_m(i))$
6: **end for**
*Sort the **feature** array in ascending order of positional weights*
7: Ranks = *sort*(feature)
8: **return** *Ranks*
/ **pos(r_j(i))** defines the position of i^{th} feature in r_j ranking */*

3.2.4 One-class classification approach

From the feature vector obtained as a result of rank aggregation, top '*r*' features are used to train the corresponding classifier. Amongst the limited works deploying one class classification approaches for the task, the classifiers namely, Local Outlier Factor (LOF), IsolationForest and One-Class SVM (OSVM) are found to perform well for text classification as well as authorship verification problems (Neal et al. 2018; Jankowska et al. 2013; Koppel and Schler 2004). Hence, efficiency of the stated classifiers has been analyzed in this work. One class classifiers are mainly deployed for the anomaly or novelty detection tasks where patterns from the normal training samples are learnt and any data sample lying outside this learned pattern is classified as anomalous. In other words, from only the normal class training data, an algorithm builds a representational model. In case any newly found data is observed to be different from the trained model, the same is considered as anomalous. In the authorship verification problem also, the task is to learn the behavioral patterns of a user and any pattern violating the existing learned behavior need to be considered as anomalous and thus marked as a point of compromise.

Local Outlier Factor LOF (Breunig et al. 2000) is a popular unary classification algorithm mainly used for detecting anomalies in large databases. It adopts a nearest neighbor approach by comparing the distance between a text document v_i and its k-nearest neighbors to the distance between these neighbors and their respective k-nearest neighbors. Selection of parameter k plays a vital role as it represents the local neighborhood density to be considered for evaluation. Deviation of the local density of the data point from the density of its local neighbors reflects an outlier score, hence, termed as Local Outlier Factor (LOF). The neighborhood of a data point v_i is constructed using a neighborhood boundary distance d_{bd} of v_i that defines the distance d of v_i form its kth neighbor $N(v_i, k)$.

$$d_{\text{bd}}(v_i, k) = d(v_i, N(v_i, k)) \tag{16}$$

Using this boundary distance, a neighborhood $\text{NB}(v_i, k)$ is created which contains all such data points v_j whose distance to v_i is less than neighborhood boundary distance d_{bd}.

$$\text{NB}(v_i, k) = \{v_j \in X_{\text{train}} \backslash \{v_i\} \mid d(v_i, v_j) \leq d_{\text{bd}}(v_i, k)\} \tag{17}$$

Further, the density of the constructed neighborhood $\text{NB}(v_i, k)$, is computed using the reachability distance d_{rb}. As defined in Eq. (17) v_i is excluded from the neighbor space while calculating the Neighborhood $\text{NB}(v_i, k)$ and hence it is ensured using reachability distance that a minimum distance between concerned points v_i and v_j is maintained.

$$d_{\text{rb}}(v_i, v_j, k) = \max \left\{ d_{\text{bd}}(v_j, k), d(v_j, v_i) \right\} \tag{18}$$

Neighborhood data points $\text{NB}(v_i, k)$ along with their reachability distance help calculate neighborhood density D of v_i as follows:

$$D(v_i, k) = \frac{\text{NB}(v_i, k)}{\sum_{v_j \in \text{NB}(v_i, k)} d_{\text{rb}}(v_i, v_j, k)} \tag{19}$$

Next step involves the comparison of neighborhood density $D(v_i, k)$ of the data point v_i with that of the corresponding neighborhood. This help produce an anomalous score termed as Local Outlier Factor (LOF) computed as follows:

$$\text{LOF}(v_i, k) = \frac{\sum_{v_j \in \text{NB}(v_i, k)} \frac{D(v_j, k)}{D(v_i, k)}}{|\text{NB}(v_i, k)|} \tag{20}$$

A data point lying outside a cluster (i.e., a tweet sample not belonging to the user) is found to have a lower neighborhood density and hence a high LOF score and hence can be easily categorized as anomalous.

Isolation forest Isolation Forest (Liu et al. 2008) is also a (unary) one-class classifier popularly deployed for the one-class classification task. Its working is similar to its counterpart Random Forest. In Isolation Forest, like any tree ensemble method, feature space is recursively separated by creating numerous (binary) trees. As the name implies, IF isolates the data points using a randomly selected feature and a threshold split value taken as a randomly chosen value between the minimum and maximum value of that feature. Representing the split as a tree structure, total number

of splits needed to isolate the data points is based on the path length from root to the terminating leaf node. Value of an anomalous node is obviously different from the existing normal samples of a user and using random partitioning whenever any such anomalous value occurs it is observed to have shorter path length with only a small number of splits and is found near to the root node. As anomalous data (i.e., data not belonging to the same user) is different from the data samples of the concerned user, hence, such data are expected to be more prone toward isolation. Anomalous score in Isolation Forest is calculated as follows:

$$\mathrm{IF}_{\mathrm{score}}(i, n) = 2^{-\frac{E(p(i))}{q(n)}} \tag{21}$$

where for n number of nodes, $p(i)$ denotes the path length of the observation i and $q(n)$ defines the average path length of an unsuccessful search. $E(p(i))$ is an average of $p(i)$ from different isolation trees. For anomalous instances, $\mathrm{IF}_{\mathrm{score}}$ comes out to be higher, i.e., near to 1 whereas for normal instances lower scores are seen. Unlike other distance and density-based methods, isolation property of Isolation Forest aids in building partial models. Also, Isolation Forest do not require any distance or density metric, hence, it is a computationally efficient approach.

One-Class SVM One class Support Vector Machine (Schölkopf et al. 2000) is another unary classification algorithm which works on the idea of hypersphere decision boundary construction. It is often used in problems where after learning the normal behavioral pattern, task is to determine whether the new upcoming data are similar to the training data or not. This is performed by fitting a hyper sphere which includes the positive training data points. In One class SVM (OCC-SVM), the input data points are mapped using a kernel to a higher-dimensional feature space. A maximal margin hypersphere that effectively separates the training samples from origin is found. OCC-SVM is often studied as a binary class SVM with all the training data points lying in the first class and origin taken as the only data sample in the second class. Then slowly the algorithm work toward the feature space considering not only origin but the points close to origin also as outliers. The hyper sphere decision boundary adheres to the following classification rule:

$$f(x) = \langle w, x \rangle + b \tag{22}$$

where w defines a normal feature vector and b corresponds to a bias term. With an OCC-SVM, the target is to find such a rule f that maximizes the geometric margin and hence can be deployed to label a data sample x. Negative values of $f(x)$ correspond to an anomaly, i.e., for $f(x) < 0$ the behavior is labeled as anomalous, otherwise it is considered normal. For appropriate One-Class SVM working, it is optimal to chose right set of features, appropriate kernel and their respective parameters. Features present the behavioral pattern representing the data. Kernels on the other hand are used to handle the data that is not linearly separable and hence map such a data into higher dimensions. Most commonly deployed kernels include, Linear, Polynomial, Gaussian (RBF) and Sigmoid.

Working principle of OCC-SVM can be abridged as transforming the training data into another feature space S using a relevant kernel function. A maximum margin hyperplane is found that separates the transformed mapped feature vector from the origin. For $x_1, x_2, x_3 \ldots x_m$ training samples each representing a N-dimensional feature vector represented as K^N, a mapping $g(.) : K^N \mapsto S$ maps the data to a higher-dimensional feature space, S. Hereinafter, these notations have been used: the mapped data $g(x_j)$ as g_j, data points $\rho \in S$ and $\omega \in K$. In a One class SVM, the objective is to find such a hyperplane that takes most of the data points on the positive side, i.e., $\langle \rho, g_j \rangle - \omega > 0$. This is achieved by maximizing the minimal Euclidean distance $\frac{\omega}{\|\rho\|}$ between origin and the target data.

Also, in the One-Class SVM setting, a slack variable is used to minimize the error introduced by the certain preordained abnormal or noise points present in the data.

Finally, the purpose of the OCC-SVM is to handle the following optimization task:

$$\begin{aligned} &\min \frac{1}{2} \| \rho \|^2 + \frac{1}{vm} \sum_{j=1}^{m} \xi - \omega \\ &\textbf{s.t.} \ \langle \rho, g_j \rangle \leq \omega - \xi, \quad \xi \leq 0, \quad j = 1, 2, 3, \ldots m \end{aligned} \tag{23}$$

where $x_j \in K^N$ is the actual data sample and $\rho \in K$ denotes the normal vector of hyper sphere. Other parameters such as $\omega \in K$ represents the offset value; $v \in (0, 1]$, a predefined positive integer and m the total number of data points. The slack function $\frac{1}{vm} \sum_{j=1}^{m} \xi$ helps diminish the impact of abnormal points. Pattern of a data point belonging to the negative direction of hyper sphere, i.e., $\langle \rho^*, g_j \rangle + \omega^* < 0$ is often different from the given positive data points. The optimal solution generated for Eq. (23) is denoted as (ρ^*, ω^*).

4 Experimental results and discussion

This section discusses the experiments performed on the extracted features, application of feature selection, rank aggregation and then the corresponding one class classification approach to evaluate the efficiency. All the experiments are coded in Python and the machine learning toolkit called scikit-learn (Pedregosa et al. 2011) has been used for classification.

4.1 Data collection

Basic prerequisite before performing any experiment is the collection/creation of data set. Among various prevalent social networks, Twitter has always been the gold standard chosen by researchers for experiments because of the accessibility and flexibility of API services. In the literature, time ordered data of a large set of Twitter users (approximately, 147K users) has been collected in Li et al. (2012) but it is limited to the availability of only 500 tweets of each user. Furthermore, the collected tweets are from the year 2012. Though approach designed in this work is a generalized one and is independent of the time and source of data, still in the subsequent years, Twitter as a platform has undergone various modifications such as increase of tweet size from 140 to 280 characters, introduction of media tags, image and video links not counted as character limit, enabling retweets on own tweets etc. In the said dataset, the number of tweets collected from each user are less and insufficient which may limit the amount of data for profiling and train-test sampling, hence, it was preferred to conduct experiments with a comparatively large amount of latest tweets to strengthen the stated claims. Usernames of a set of 4000 users are randomly chosen from the available 147K users and 4000 latest tweets for each user were extracted using the Twitter API. Average length of each tweet in the dataset is observed to be 127 characters.

Users with less than 4000 tweets are discarded which left us with 3057 users for experiments. The count of users and tweets is made taking into consideration the availability of resources for conducting experiments. But this nowhere limits the approach on these parameters as each user is independently analyzed hence, experiments can be performed with any number of users.

4.2 Data preprocessing

Because of the limitation of 140/280 characters of tweets, two tweets are combined to form a slightly larger text for easy and fair analysis. Also, tweets involving Retweets and UNICODE blocks are removed. For stylometric features, no other preprocessing is performed as presence of every minor detail has an influential impact on performance. But for other feature analysis, preprocessing steps involve conversion of each tweet in lower case followed by removal of punctuation and stopwords.

4.3 Experimental approach

Varying the hyper parameters of classifiers, 76 experiments are performed. Experiments are performed on a 64-bit Windows Operating system with 4.2 GHz i7 processor and 32GB RAM. A total of 4000 tweets are collected for each user and combining two tweets to form a larger text leaves resultant 2000 tweets for experiments. Amongst these 2000 tweets acting as positive class data, 50% (i.e., 1000) tweets are used for creating a baseline set and remaining 50% (i.e., 1000) tweets are reserved to be used as train (60%) and test (40%) set for performance evaluation. Interspersing is used to disperse the data into respective sets. A Shuffle-Split approach with 5-fold cross-validation is performed on training data (split into 60–40 train and validation set). Same procedure is repeated for each user. Unlike, binary or multi-class classification problems, in one class classification, respective classifiers are trained only with the positive class data. But the test data are undoubtedly induced with some random negative samples in order to analyze the efficiency of classifiers for both the scenarios, i.e., true positives and true negatives to judge how efficient the model is in recognizing the respective normal and abnormal data points. Every classifier is parameter-tuned and trained with the feature vector obtained from the top r features from BORDA technique. Top Ranked features selected using each independent feature selection technique as well as BORDA rank aggregation is shown in Fig. 2. As each feature selection technique has a different policy of selecting features, hence, it lead to the selection of different features. To overcome this ambiguity, aggregation of features is performed using BORDA and the ranks generated using BORDA are considered for further experiments. After generating the ranks, it is usually practiced to select top r features and perform experiments on them. Keeping computational time as well as resource constraints into consideration, to decide for the value of r, feature importance experiments have been performed on the data of randomly chosen 500 users by varying the value of r and choosing top r features every time for the experiments. Performance efficiency of three one-class classifiers namely, LOF, IF and OCC-SVM has been evaluated on the basis of various performance measures mentioned in Table 2.

4.4 Results and discussion

The driving motivation for this paper is to analyze the efficacy of various textual features and one class classification algorithms for the task of authorship verification. The task is to asses which features and algorithm as well as what set of fine tuned parameters help achieve better results. Figures 3, 4, 5, 6, 7 and 8 present the detailed variation in behavior of different classifiers on change of parameter settings. Secondly, it also signifies how selection of top ranked features influences the performance of the classifiers. From Fig. 3, it has been observed that for LOF as a classifier, with the increase in number of features, performance increases marginally but becomes almost stable after the inclusion of sufficient number of top ranked features (i.e., 15 features). Further inclusion of features deteriorates the performance. Similarly, in the case of Isolation Forest (Fig. 4), continu-

Table 2 Performance evaluation metrics (Sokolova and Lapalme 2009)

Metric	Definition	Formula	Remarks
Accuracy	Accuracy specifies the ratio of correctly recognized instances to the total number of instances	$\text{Acc} = \frac{\text{Correctly recognized tweets}}{\text{Total number of tweets}}$	Not an efficient measure for imbalanced datasets. In that case, if the model randomly guess the data points as belonging to the majority class, a very high accuracy may be obtained although classifier has not learnt anything
Precision	A performance metric to compute correctly recognized tweets of an authentic user using true positive and false positive rate	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$	High precision—Higher recognition of tweets of the same user. Low precision—High amount of other users' tweets incorrectly recognized as that of the said user
Recall	A performance metric that accounts the correct classification of tweets by an outsider using both true positives and false negatives	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$	Higher recall value signifies high proportion of tweets correctly recognized by outside user. Lower recall value detects high amount of incorrect recognition of tweets of legitimate users
F-score	F-score as performance measure gives equal importance to precision and recall computing a harmonic mean of both the measures	$F\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Higher values—better model. Although an effective measure, but it does not take true negatives into account, hence may be superseded by other measures for better performance analysis
Matthews Correlation Coefficient (M_{cc})	A balanced performance measure to analyze the effectiveness of a classification model taking into account all the four confusion matrix quadrants namely, true positives, false positives, false negatives and true negatives	$\frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TN} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$	M_{cc} values range from -1 to $+1$ and higher values signify better classification model with matching predicted and original output values. Lower values toward negative range indicate a wrong classification model with completely opposite prediction
Zero_One_Loss	A loss function which counts the misclassification of labels. In its actual form it returns the total number of instances (tweets) misclassified but in the normalized form it computes the proportion of instances (tweets) incorrectly recognized w.r.t the total number of instances (tweets)	$\frac{(\text{FP} + \text{FN})}{\text{Total number of tweets}}$	Low values (i.e. 0) signify the correct classification with 0 losses and higher values (1) signify incorrect classification with all the losses
False rejection rate (FRR)	FRR depicts the rate of incorrectly recognizing a genuine user as an imposter and is calculated as the ratio of false negatives to the total number of instances (tweets)	$\text{FRR} = \frac{\text{FN}}{\text{FN} + \text{TP}}$	Signifies the likelihood of a genuine user getting rejected. Also, lower values of FRR are preferred
False acceptance rate (FAR)	FAR depicts the rate of incorrectly recognizing an imposter as a genuine user and is calculated as the ratio of false positives to the total number of instances (tweets)	$\text{FAR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$	Signifies the likelihood of a unauthentic user getting accepted. Like FRR, lower values for FAR are preferred

TP true positives, *FP* false positives, *FN* false negatives, *TN* true negatives

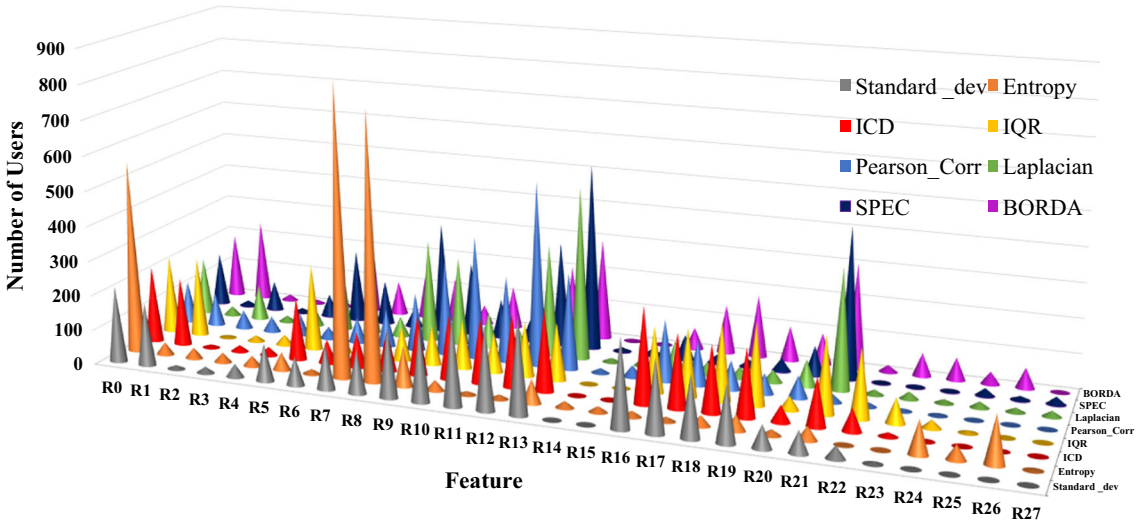


Fig. 2 Top ranked features using different feature selection techniques and BORDA in terms of count of users

ous addition of features increases the performance but only up to a certain limit (till $r = 9$ or 10). It is observed that addition of features beyond that point decreases the performance. Likewise, under varying parameter settings and kernels in OCC-SVM, performance initially improve but further deteriorate beyond addition of r features (Figs. 5, 6, 7, 8). The value of r beyond which performance of classifiers deteriorates has been seen to be top 15, 9 and 11 features for ***LOF (n_neighbors = 9)*** *(Accuracy: 72.56%, Precision: 84.86%, Recall: 71.52%, F-score: 76.28%, Matthews Correlation Coefficient: 47.28%, Zero_one_loss: 27.43%, FRR:28.48%, FAR: 21.73%),* ***IF (n_estimators = 100)*** *(Accuracy: 77.86%, Precision: 85.04%, Recall: 77.44%, F-score: 79.92%, Matthews Correlation Coefficient: 57.44%, Zero_one_loss: 22.14%, FRR: 22.56%, FAR: 18.10%) and* ***OCC-SVM (Gamma = 1,nu = 0.1,kernel = rbf)*** *(Accuracy: 86.94%, Precision: 88.49%, Recall:87.91%, F-score: 87.44%, Matthews Correlation Coefficient: 75.02%, Zero_one_loss:13.06%, FRR: 12.08%, FAR:11.53%),* respectively.

- From the experimental analysis, it is seen that increase in $n\_^{neighbors}$ in LOF improves the performance (Fig. 3). Experiments are performed varying the values of n_neighbors(K) from 3 to 9 and F-score difference of around 3.13% is seen in results using $top15$ features.

- Similarly, using Isolation Forest as a one class classifier it is found that accuracy slightly improved on increasing n_estimators (Fig. 4) but on the cost of increased computational time for training and validation. The increase in n_estimators added only a negligible difference in performance improvement (1.05% F-score) with $top9$ features. It is to be noted that the default contamination parameter value in Isolation Forest is 0.1, i.e., in the training examples the model has adjusted itself to at least have 10% anomalies which are the data points not resembling remaining normal points. This value has explicitly been set to 0 so as to avoid model getting adjusted to the contaminated values.

- Likewise, in OCC-SVM, $gamma$, nu and $kernel$ parameters are varied and it is observed that the choice of kernel complimented with the nu and $gamma$ values impact the performance. As evident from Figs. 5, 6, 7 and 8, higher values of nu deteriorates the performance. Hence, with all the $kernels$, $nu = 0.1$ gave better performance than other values. Amongst different kernels, rbf kernel with $Gamma = 1$ gave 2.22%, 3.22%, 5.51% better performance than its $linear$, $poly$ and $sigmoid$ counterparts at their respective best parameter settings. Overall rbf kernel with nu and $Gamma$ values set to 0.1 and 1, respectively, attained 87.44% F-score which is better than the other combinations.

With the top r features, once the parameter-tuned classifier is trained, test data is fed to each parameter-tuned classifier to validate the model and obtain the evaluation metrics. Performance of the discussed classifiers under optimal parameter settings have been analyzed on the basis of various evaluation metrics and is shown in Table 3. It is evident that OCC-SVM outperformed other one-class classifiers in terms of different performance parameters. Moreover, OCC-SVM with rbf kernel took lowest time to train.

Tabulated values represent the average scores of all users on their test sets. Every user considered for experiments has different set of negative samples, but the random placement of tweets in each user's sample allows us to use any such

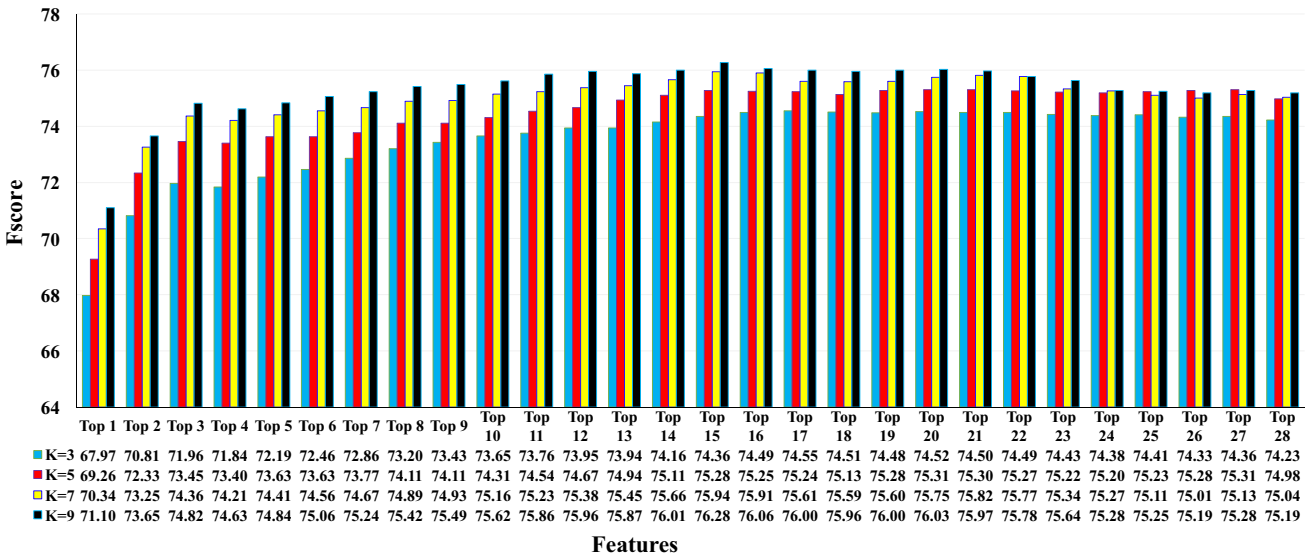


Fig. 3 Performance analysis of *LOF* (*with varying* $n_{neighbors}$) by taking top '*k*' features from BORDA

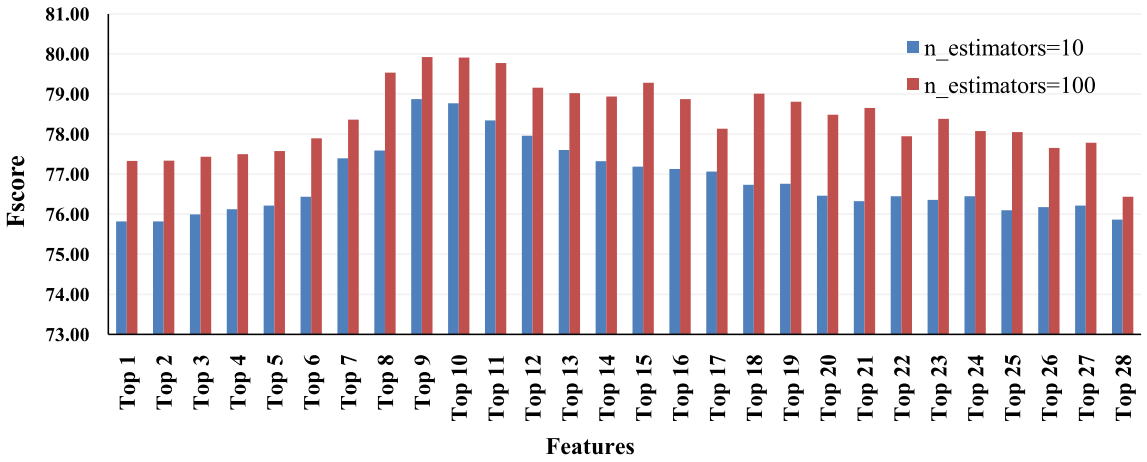


Fig. 4 Performance analysis of *Isolation Forest* (*with varying* $n_{estimators}$ by taking top '*k*' features from BORDA

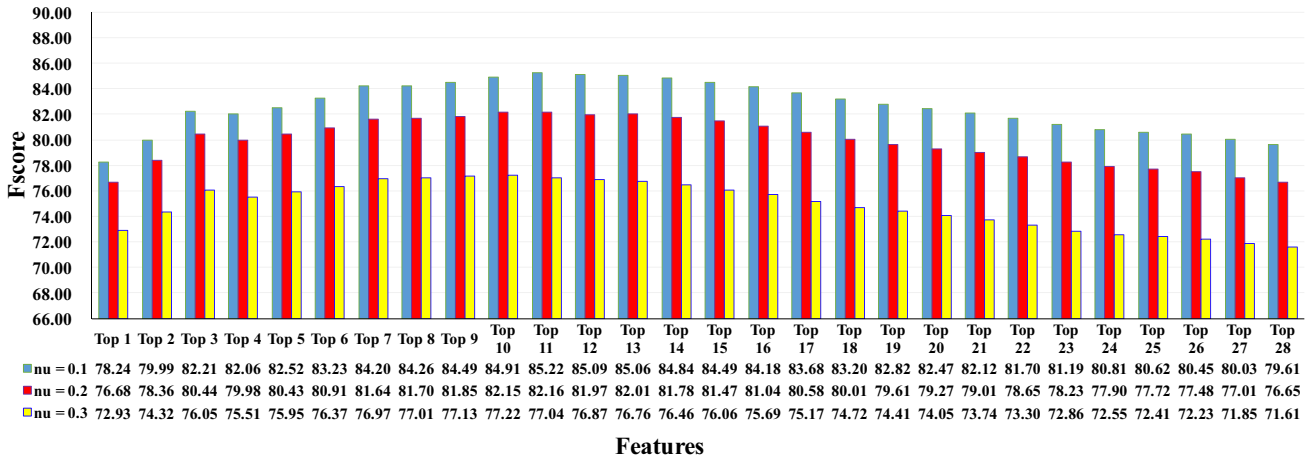
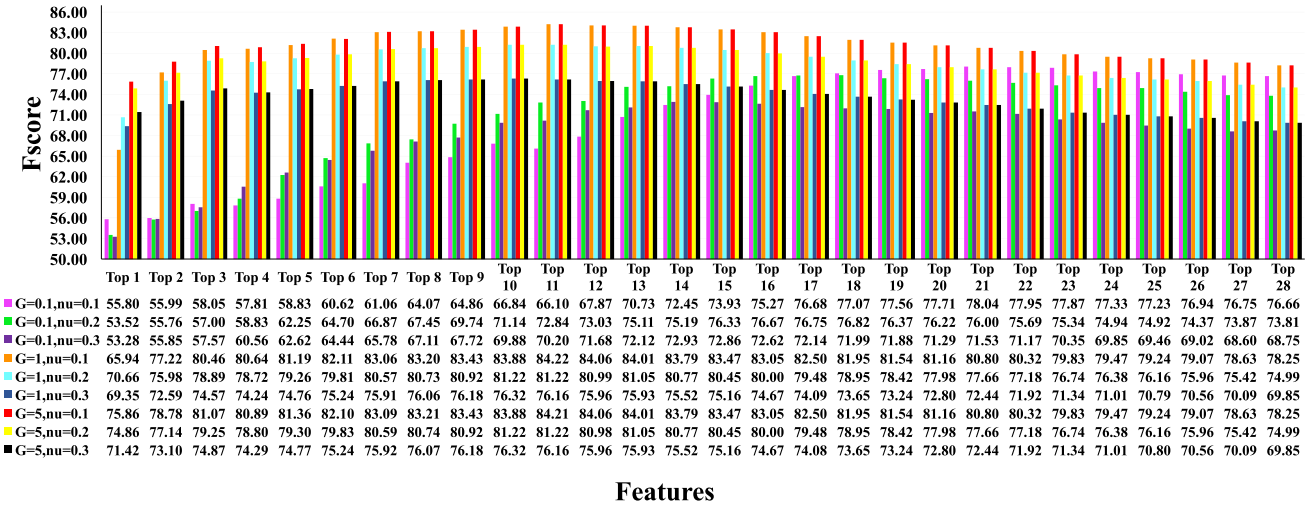
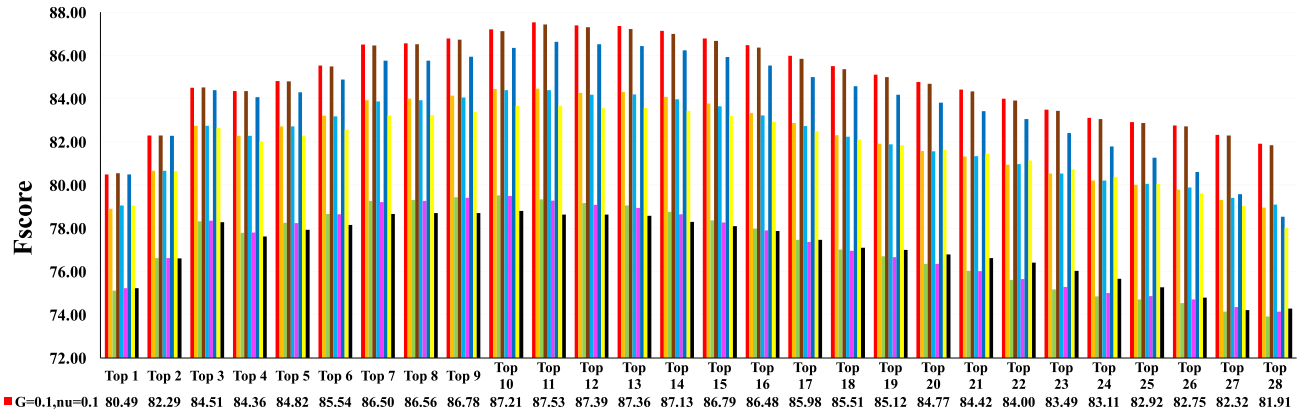


Fig. 5 Performance analysis of *Linear Kernel of OCC-SVM (with varying nu parameter)* by taking top '*k*' features from BORDA



	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7	Top 8	Top 9	Top 10	Top 11	Top 12	Top 13	Top 14	Top 15	Top 16	Top 17	Top 18	Top 19	Top 20	Top 21	Top 22	Top 23	Top 24	Top 25	Top 26	Top 27	Top 28
G=0.1,nu=0.1	55.80	55.99	58.05	57.81	58.83	60.62	61.06	64.07	64.86	66.84	66.10	67.87	70.73	72.45	73.93	75.27	76.68	77.07	77.56	77.71	78.04	77.95	77.87	77.33	77.23	76.94	76.75	76.66
G=0.1,nu=0.2	53.52	55.76	57.00	58.83	62.25	64.70	66.87	67.45	69.74	71.14	72.84	73.03	75.11	75.19	76.33	76.67	76.75	76.82	76.37	76.22	76.00	75.69	75.34	74.94	74.92	74.37	73.87	73.81
G=0.1,nu=0.3	53.28	55.85	57.57	60.56	62.62	64.44	65.78	67.11	67.72	69.88	70.20	71.68	72.12	72.93	72.86	72.62	72.14	71.99	71.88	71.29	71.53	71.17	70.35	69.85	69.46	69.02	68.60	68.75
G=1,nu=0.1	65.94	77.22	80.46	80.64	81.19	82.11	83.06	83.20	83.43	83.88	84.22	84.06	84.01	83.79	83.47	83.05	82.50	81.95	81.54	81.16	80.80	80.32	79.83	79.47	79.24	79.07	78.63	78.25
G=1,nu=0.2	70.66	75.98	78.89	78.72	79.26	79.81	80.57	80.73	80.92	81.22	81.22	80.99	81.05	80.77	80.45	80.00	79.48	78.95	78.42	77.98	77.66	77.18	76.74	76.38	76.16	75.96	75.42	74.99
G=1,nu=0.3	69.35	72.59	74.57	74.24	74.76	75.24	75.91	76.06	76.18	76.32	76.16	75.96	75.93	75.52	75.16	74.67	74.09	73.65	73.24	72.80	72.44	71.92	71.34	71.01	70.79	70.56	70.09	69.85
G=5,nu=0.1	75.86	78.78	81.07	80.89	81.36	82.10	83.09	83.21	83.43	83.88	84.21	84.06	84.01	83.79	83.47	83.05	82.50	81.95	81.54	81.16	80.80	80.32	79.83	79.47	79.24	79.07	78.63	78.25
G=5,nu=0.2	74.86	77.14	79.25	78.80	79.30	79.83	80.59	80.74	80.92	81.22	81.22	80.98	81.05	80.77	80.45	80.00	79.48	78.95	78.42	77.98	77.66	77.18	76.74	76.38	76.16	75.96	75.42	74.99
G=5,nu=0.3	71.42	73.10	74.87	74.29	74.77	75.24	75.92	76.07	76.18	76.32	76.16	75.96	75.93	75.52	75.16	74.67	74.08	73.65	73.24	72.80	72.44	71.92	71.34	71.01	70.80	70.56	70.09	69.85

Fig. 6 Performance analysis of *Polynomial Kernel of OCC-SVM* (*varying Gamma and nu parameter*) by taking top '*k*' features from BORDA



	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7	Top 8	Top 9	Top 10	Top 11	Top 12	Top 13	Top 14	Top 15	Top 16	Top 17	Top 18	Top 19	Top 20	Top 21	Top 22	Top 23	Top 24	Top 25	Top 26	Top 27	Top 28
G=0.1,nu=0.1	80.49	82.29	84.51	84.36	84.82	85.54	86.50	86.56	86.78	87.21	87.53	87.39	87.36	87.13	86.79	86.48	85.98	85.51	85.12	84.77	84.42	84.00	83.49	83.11	82.92	82.75	82.32	81.91
G=0.1,nu=0.2	78.91	80.66	82.74	82.28	82.72	83.21	83.92	84.00	84.14	84.45	84.46	84.27	84.31	84.08	83.77	83.34	82.87	82.31	81.91	81.57	81.32	80.95	80.53	80.20	80.02	79.78	79.32	78.95
G=0.1,nu=0.3	75.11	76.62	78.33	77.79	78.25	78.66	79.27	79.31	79.43	79.52	79.34	79.17	79.05	78.76	78.36	77.99	77.46	77.02	76.70	76.35	76.04	75.60	75.16	74.85	74.71	74.54	74.14	73.91
G=1,nu=0.1	80.56	82.29	84.51	84.35	84.79	85.49	86.46	86.52	86.72	87.12	87.44	87.31	87.22	86.99	86.68	86.36	85.85	85.37	84.99	84.68	84.34	83.91	83.43	83.05	82.87	82.72	82.29	81.85
G=1,nu=0.2	79.05	80.66	82.74	82.28	82.72	83.18	83.88	83.93	84.06	84.39	84.39	84.18	84.19	83.97	83.64	83.22	82.73	82.24	81.88	81.56	81.34	80.97	80.54	80.22	80.06	79.88	79.40	79.10
G=1,nu=0.3	75.23	76.61	78.35	77.80	78.24	78.64	79.21	79.26	79.41	79.49	79.28	79.09	78.94	78.65	78.26	77.90	77.37	76.96	76.66	76.35	76.01	75.65	75.29	75.00	74.86	74.70	74.35	74.14
G=5,nu=0.1	80.49	82.29	84.40	84.07	84.30	84.89	85.75	85.76	85.95	86.35	86.63	86.51	86.43	86.24	85.92	85.53	85.00	84.58	84.18	83.82	83.42	83.06	82.40	81.79	81.27	80.60	79.58	78.54
G=5,nu=0.2	79.05	80.65	82.65	82.03	82.28	82.57	83.22	83.24	83.40	83.67	83.67	83.56	83.58	83.44	83.20	82.93	82.49	82.12	81.85	81.63	81.45	81.14	80.73	80.38	80.06	79.62	79.03	78.03
G=5,nu=0.3	75.23	76.60	78.29	77.62	77.93	78.15	78.67	78.70	78.70	78.81	78.64	78.63	78.58	78.29	78.10	77.87	77.47	77.09	77.00	76.79	76.62	76.40	76.03	75.67	75.27	74.80	74.21	74.28

Fig. 7 Performance analysis of *rbf Kernel of OCC-SVM* (*with varying Gamma and nu parameter)* by taking top '*k*' features from BORDA

sample. Still in order to avoid the ambiguity, during testing phase, we performed 100 iterations of experiments for each user taking same positive samples but random negative samples in each iteration. Figure 9 represents the deviation in performance (in terms of coefficient of variance, CV of *F*-score) for each user over the 100 iterations. coefficient of variance (CV) signifies the relative variability in values. It is computed using the standard deviation and mean values of the respective performance metric. CV values obtained by different users over 100 iterations have been sorted and then plotted to avoid a messy graph.

In order to demonstrate the performance of each fine tuned one-class classifier for all the users, a boxplot (Fig. 10) is shown with 25 to 75% interquartile range. It is seen that for each classifier most of the values skewed to the higher *F*-score range having a median *F*-score of 75%, 81% and 82% for *LOF, IF* and *OCC-SVM*, respectively. For some 25% users, *F*-score values ranged from lower quartile (approx 70%, 71%, 75%) to the minimum whisker of 52%, 46% and 52%, respectively. Only a small amount of users obtained *F*-score less than the corresponding minimum whisker values.

4.5 Comparison with existing authorship verification approaches

In order to compare the performance of the proposed method against existing authorship verification methods, three baseline techniques (Brocardo et al. 2015; Barbon et al. 2017;

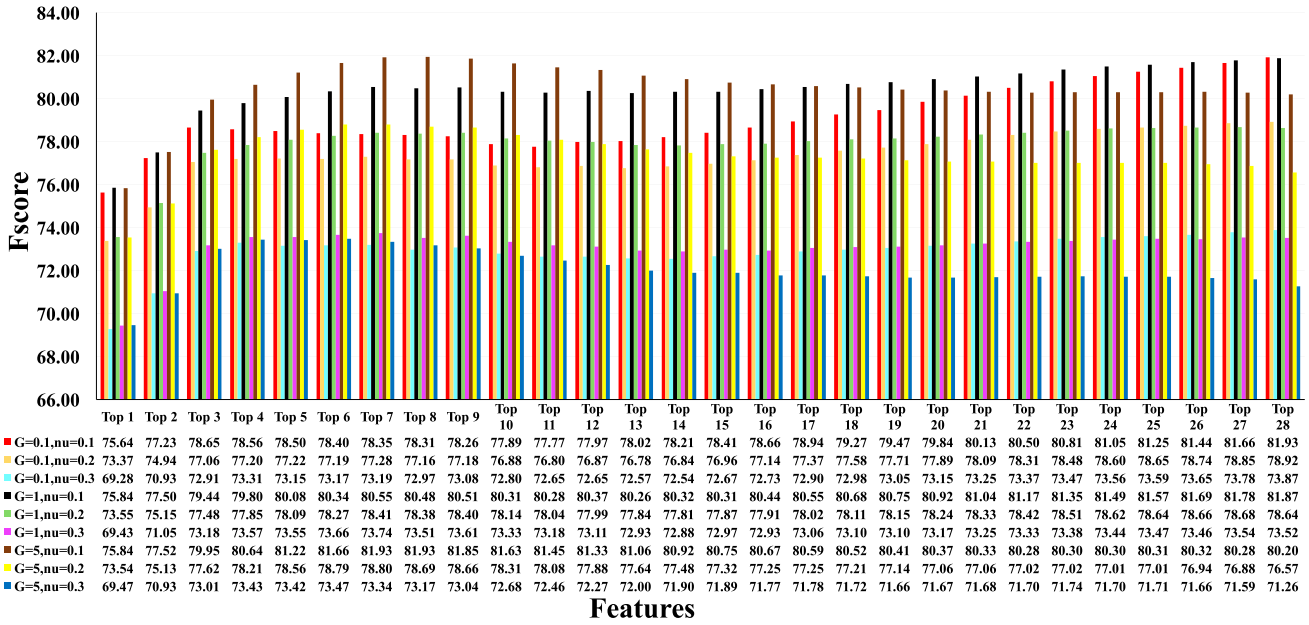


Fig. 8 Performance analysis of *Sigmoid Kernel of OCC-SVM* (*with varying Gamma and nu parameter)* by taking top '*k*' features from BORDA

Table 3 Performance of different one-class classifiers with their optimized parameters

Classifier	LOF	IF	OCC-SVM
Optimized parameters	*n*_neighbors = [1,3,5,9]	*n*_estimators = [10,100]	Gamma = [0.1,1,5,10], nu = [0.1,0.2,0.3], kernel=['Linear','poly', 'rbf','sigmoid']
Best Parameter Setting	*n*_neighbors = 9	*n*_estimators = 100	Gamma = 1, nu = 0.1, kernel = 'rbf'
Accuracy (%)	72.55	77.57	86.61
Precision (%)	84.74	84.96	88.93
Recall (%)	71.57	77.26	87.20
F-score (%)	76.15	79.66	87.29
Matthews correlation coefficient (%)	47.42	56.90	74.42
Zero_one_loss (%)	27.45	22.43	13.39
FRR (%)	28.42	22.74	12.79
FAR (%)	21.39	18.18	11.33

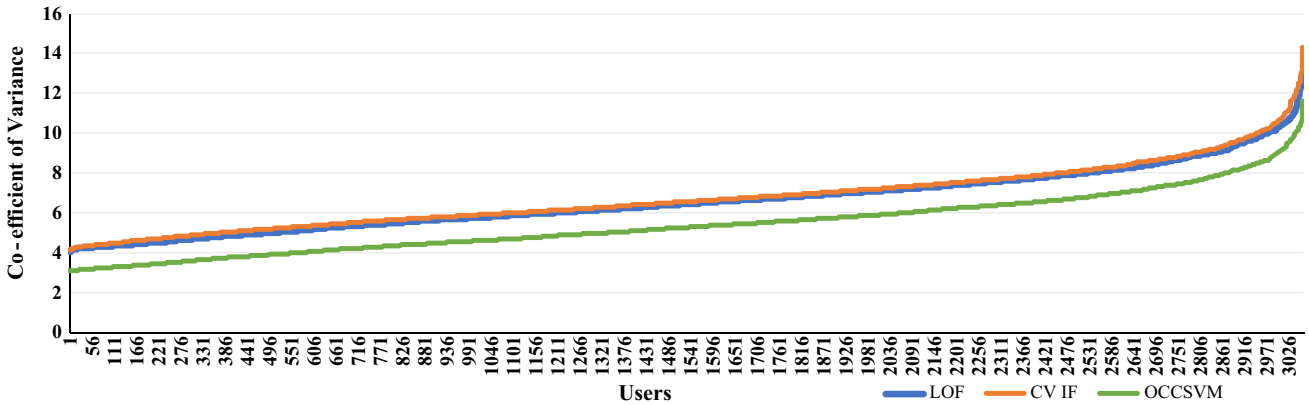


Fig. 9 Deviation in *F*-score performance by One class classifiers over 100 iterations of varied test samples

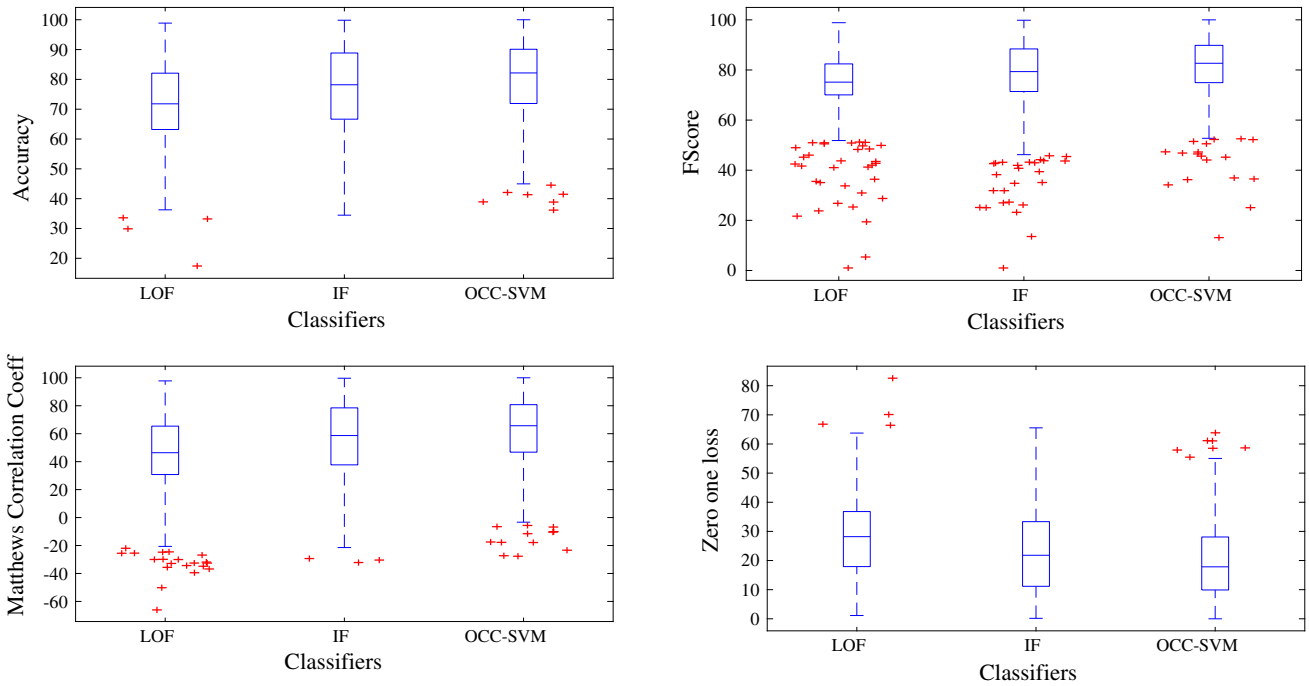


Fig. 10 Boxplots representing variation of different evaluation metrics with different classifiers considering all the 3057 users

Li et al. 2016) which have also worked on the detection of compromised accounts in social networks using textual features have been studied and implemented. These existing methods have studied the problem from binary classification perspective considering data samples from other users as negative class data. Though binary class AV methods achieve better performance than unary methods but the former limits the problem for real time deployment as the models are trained with a limited amount of negative (anomalous data). Also, with this limited negative class data, any new abnormal behavior not previously learned in the training phase would not be correctly detected at run time. On the other hand, unary methods are only trained with the positive class data and anything lying outside this positive learned pattern is considered as anomalous. Hence, it is more appropriate and reliable to deploy unary models in real environment. For a fair comparison, features and methods used in these existing techniques have been deployed on the collected Twitter dataset of 3057 users. Table 4 presents the tabulated performance comparison of the proposed intrinsic technique against the existing techniques.

From Fig. 11 it is evident that the proposed intrinsic technique has a comparable performance to other existing authorship verification techniques. Being practically more reliable and authentic, it is better to study authorship verification as a one-class classification problem than a binary classification problem. Moreover, even the unary classifiers attains at par performance with binary ones, hence it is viable to use unary models for the task.

5 Practical applicability of the proposed approach

Other ways for checking the authenticity of a user and thereby detecting compromised accounts in social networks include the use of mouse dynamics, keyboard strokes, physiological biometrics, face/iris/fingerprint recognition, behavioral biometrics and many others. But these physical approaches are too costly to be deployed as they require specialized hardware support. Moreover, once forged or compromised it becomes extremely difficult to replace them. On the other hand, authorship verification task can be applied unobtrusively at the back end without the active participation of the concerned user. The required information could be gathered smoothly at the service provider's end so as to not have the unnecessary active involvement of a user every time a check needs to be done. Also, no specially designed hardware is required for the profiling-based authorship analysis task. As extra security is always a good idea, hence verification of tweets could be deployed continuously at the back end as a complementary measure to the other compromised account detection approaches. Furthermore, the proposed technique is applicable over different literary works as well as forensic studies to check the authenticity of the text. This will definitely aid security personnel in the digital forensic scenarios.

Table 4 Comparative analysis of the proposed intrinsic technique with existing techniques

Technique	Features	Classifier	Accuracy (Avg) (%)	Precision (Avg) (%)	Recall (Avg) (%)	*F*-score (Avg) (%)
Proposed intrinsic technique	Content-specific and Content-free	OCC-SVM with 'rbf' kernel	86.61	88.93	87.20	87.29
Barbon et al. (2017)	char *n*-grams (SPI)	kNN	84.75	85.38	85.64	86.27
Li et al. (2016)	Stylometric	Ensemble Voting	84.57	84.24	84.87	84.52
Li et al. (2016)	Stylometric	Decision Trees	70.58	70.35	70.59	70.43
Brocardo et al. (2015)	*n*-grams	Hybrid SVM-LR	89.57	89.59	89.54	89.53

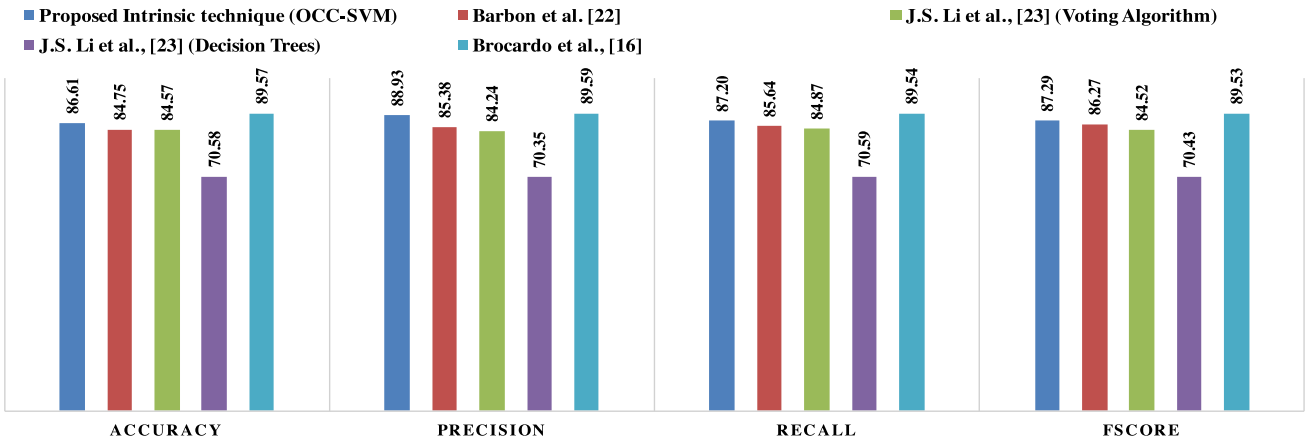


Fig. 11 Comparison of the proposed technique with other existing techniques in terms of various evaluation metrics

6 Conclusion

In this work, efficiency of different textual features for the task of authorship verification and thereby the detection of compromised accounts in social networks have been examined. Experiments have been performed on tweets of a popular social network Twitter but it nowhere makes the proposed method specific to Twitter platform, indeed it could be deployed on any social network. Various feature selection techniques have been used to rank the features which are then aggregated using a popular rank aggregation technique called BORDA. The problem of authorship verification has been studied in its actual unary form where only the data samples from the same user have been used for profiling and training of models. Efficiency of various one-class classifiers is examined with experimental results stating that One Class SVM with rbf kernal outperformed other classifiers namely, LOF and Isolation Forest, attaining an average *F*-score of 87.29% and Matthews Correlation Coefficient of 74.42% under varied parameter settings. Unlike binary classification where data samples in the negative class are randomly collected to train the models, reliability on unary models seems more viable as they are trained using only the positive class, i.e., data samples of the respective user. Thus studying the authorship verification problem from unary classification perspective seems more reasonable. The work undertaken is grounded only on text mining, hence, only text is deployed for continuous authentication which makes it an easy and unobtrusive approach to deploy at back end. In the near future, we are planning to supplement the proposed work with some meta data information such as time, language and source to compare the effectiveness of plain textual information with the other alternatives.

Acknowledgements This publication is an outcome of the R&D work under the Visvesvaraya Ph.D. Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation under the Grant No. PhD/MLA/4(61)/2015-16.

Compliance with ethical standards

Conflict of interest All the authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with direct human participants or animals performed by any of the authors. However, public tweets using Twitter API were fetched for around 3000

users. While fetching the tweets full adherence to Twitter Developer policy and agreement was made.

References

Al-Andoli M, Cheah WP, Tan SC (2020) Deep learning-based community detection in complex networks with network partitioning and reduction of trainable parameters. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02389-x

Al-Ayyoub M, Al-andoli M, Jararweh Y, Smadi M, Gupta B (2019) Improving fuzzy c-mean-based community detection in social networks using dynamic parallelism. Comput Electr Eng 74:533–546

Al-Qurishi M, Alhuzami S, AlRubaian M, Hossain MS, Alamri A, Rahman MA (2018) User profiling for big social media data using standing ovation model. Multimed Tools Appl 77(9):11179–11201

Amato F, Moscato V, Picariello A, Sperli'ì G (2019) Extreme events management using multimedia social networks. Future Gener Comput Syst 94:444–452

Barbon S, Igawa RA, Zarpelao BB (2017) Authorship verification applied to detection of compromised accounts on online social networks. Multimed Tools Appl 76(3):3213–3233

Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) Lof: identifying density-based local outliers. In: ACM sigmod record, vol 29. ACM, pp 93–104

Brocardo ML, Traore I (2014) Continuous authentication using micro-messages. In: 2014 12th annual international conference on privacy, security and trust (PST). IEEE, pp 179–188

Brocardo ML, Traore I, Woungang I (2015) Authorship verification of e-mail and tweet messages applied for continuous authentication. J Comput Syst Sci 81(8):1429–1440

Brocardo ML, Traore I, Woungang I, Obaidat MS (2017) Authorship verification using deep belief network systems. Int J Commun Syst 30(12):1–10

Brocardo ML, Traore I, Woungang I (2019) Continuous authentication using writing style. In: Obaidat M, Traore I, Woungang I (eds) Biometric-based physical and cybersecurity systems. Springer, Berlin, pp 211–232

Chakraborty M, Pal S, Pramanik R, Chowdary CR (2016) Recent developments in social spam detection and combating techniques: a survey. Inf Process Manag 52(6):1053–1073

de Borda JC (1784) Mémoire sur les élections au scrutin

Dwork C, Kumar R, Naor M, Sivakumar D (2001) Rank aggregation methods for the web. In: Proceedings of the 10th international conference on World Wide Web. ACM, pp 613–622

Egele M, Stringhini G, Kruegel C, Vigna G (2017) Towards detecting compromised accounts on social networks. IEEE Trans Dependable Secure Comput 14(4):447–460

Feng W, Zhang Z, Wang J, Han L (2016) A novel authorization delegation scheme for multimedia social networks by using proxy re-encryption. Multimed Tools Appl 75(21):13995–14014

Green RM, Sheppard JW (2013) Comparing frequency-and style-based features for twitter author identification. In: FLAIRS conference. AAAI, pp 64–69

Gupta BB, Perez GM, Agrawal DP, Gupta D (2020) Handbook of computer networks and cyber security. Springer, Berlin

Halvani O, Steinebach M (2014) Vebav-a simple, scalable and fast authorship verification scheme. In: CLEF (working notes), pp 1049–1062

Halvani O, Graner L, Vogel I (2018a) Authorship verification in the absence of explicit features and thresholds. In: European conference on information retrieval. Springer, pp 454–465

Halvani O, Winter C, Graner L (2018b) Unary and binary classification approaches and their implications for authorship verification. arXiv preprint arXiv:1901.00399

Igawa RA, Almeida A, Zarpelão B, Barbon S Jr (2016) Recognition on online social network by user's writing style. iSys-Revista Brasileira de Sistemas de Informação 8(3):64–85

Inuwa-Dutse I, Liptrott M, Korkontzelos I (2018) Detection of spam-posting accounts on twitter. Neurocomputing 315:496–511

Jankowska M, Keselj V, Milios E (2013) Proximity based one-class classification with common n-gram dissimilarity for authorship verification task. In: CLEF 2013 evaluation labs and workshop—working notes papers, pp 23–26

Javed A, Burnap P, Rana O (2019) Prediction of drive-by download attacks on twitter. Inf Process Manag 56(3):1133–1145

Kaur R, Singh S, Kumar H (2018a) Rise of spam and compromised accounts in online social networks: a state-of-the-art review of different combating approaches. J Netw Comput Appl 112:53–88

Kaur R, Singh S, Kumar H (2018b) Authcom: authorship verification and compromised account detection in online social networks using ahp-topsis embedded profiling based technique. Expert Syst Appl 113:397–414

Kocher M, Savoy J (2016) Unine at clef 2016: author profiling. In: CLEF (working notes), pp 903–911

Kocher M, Savoy J (2017) A simple and efficient algorithm for authorship verification. J Assoc Inf Sci Technol 68(1):259–269

Koppel M, Schler J (2004) Authorship verification as a one-class classification problem. In: Proceedings of the 21st international conference on machine learning. ACM, p 62

Koppel M, Winter Y (2014) Determining if two documents are written by the same author. J Assoc Inf Sci Technol 65(1):178–187

Li R, Wang S, Deng H, Wang R, Chang KC-C (2012) Towards social user profiling: unified and discriminative influence model for inferring home locations. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 1023–1031

Li JS, Chen L-C, Monaco JV, Singh P, Tappert CC (2016) A comparison of classifiers and features for authorship authentication of social networking messages. Concurr Comput Pract Exp 29(14):1–15

Li C, Zhang Z, Zhang L (2018) A novel authorization scheme for multimedia social networks under cloud storage method by using ma-cp-abe. Int J Cloud Appl Comput 8(3):32–47

Liu FT, Ting KM, Zhou Z-H (2008) Isolation forest. In: 2008 8th IEEE international conference on data mining. IEEE, pp 413–422

Lorena LH, Carvalho AC, Lorena AC (2015) Filter feature selection for one-class classification. J Intell Robot Syst 80(1):227–243

Miller Z, Dickinson B, Deitrick W, Hu W, Wang AH (2014) Twitter spammer detection using data stream clustering. Inf Sci 260:64–73

Namsrai E, Munkhdalai T, Li M, Shin J-H, Namsrai O-E, Ryu KH (2013) A feature selection-based ensemble method for arrhythmia classification. J Inf Process Syst 9(1):31–40

Neal T, Sundararajan K, Woodard D (2018) Exploiting linguistic style as a cognitive biometric for continuous verification. In: 2018 international conference on biometrics (ICB). IEEE, pp 270–276

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Peng J, Choo K-KR, Ashman H (2016) Bit-level n-gram based forensic authorship analysis on social media: identifying individuals from linguistic profiles. J Netw Comput Appl 70:171–182

Prati RC (2012) Combining feature ranking algorithms through rank aggregation. In: The 2012 international joint conference on neural networks (IJCNN). IEEE, pp 1–8

Ruan X, Wu Z, Wang H, Jajodia S (2016) Profiling online social behaviors for compromised account detection. IEEE Trans Inf Forensics Secur 11(1):176–187

Saari D (2001) Chaotic elections!: a mathematician looks at voting. American Mathematical Society, Providence
Sageder J, Demleitner A, Irlbacher O, Wimmer R (2019) Applying voting methods in user research. In: Proceedings of Mensch und computer 2019, pp 571–575
Sahoo SR, Gupta BB (2019) Hybrid approach for detection of malicious profiles in twitter. Comput Elect Eng 76:65–81
Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC (2000) Support vector method for novelty detection. Adv Neural Inf Process Syst 12:582–588
Seidman S (2013) Authorship verification using the impostors method. In: CLEF 2013 evaluation labs and workshop-online working notes, Citeseer
Serrai W, Abdelli A, Mokdad L, Hammal Y (2017) Towards an efficient and a more accurate web service selection using mcdm methods. J Comput Sci 22:253–267
Seyler D, Li L, Zhai C (2018) Identifying compromised accounts on social media using statistical text analysis. arXiv preprint arXiv:1804.07247
Shen Q, Diao R, Su P (2012) Feature selection ensemble. Turing-100 10:289–306
Singh J, Sharan A (2015) Relevance feedback based query expansion model using borda count and semantic similarity approach. Comput Intell Neurosci 2015:1–13. https://doi.org/10.1155/2015/568197
Singh M, Bansal D, Sofat S (2018) Who is who on twitter-spammer, fake or compromised account? A tool to reveal true identity in real-time. Cybern Syst 49(1):1–25
Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45(4):427–437
Stamatatos E (2009) A survey of modern authorship attribution methods. J Am Soc Inf Sci Technol 60(3):538–556
Trång D, Johansson F, Rosell M (2015) Evaluating algorithms for detection of compromised social media user accounts. In: 2nd European network intelligence conference. IEEE, pp 75–82
Tsymbal A, Pechenizkiy M, Cunningham P (2003) Diversity in ensemble feature selection. The University of Dublin: technical report TCD-CS-2003-44
Tsymbal A, Pechenizkiy M, Cunningham P (2005) Diversity in search strategies for ensemble feature selection. Inf Fusion 6(1):83–98
Van Der Walt E, Eloff J (2018) Using machine learning to detect fake identities: bots vs humans. IEEE Access 6:6540–6549
Velayudhan SP, Somasundaram MSB (2019) Compromised account detection in online social networks: a survey. Concurr Comput Practi Exp 31:e5346
Wald R, Khoshgoftaar TM, Dittman D, Awada W, Napolitano A (2012) An extensive comparison of feature ranking aggregation techniques in bioinformatics. In: 2012 IEEE 13th international conference on information reuse and integration (IRI). IEEE, pp 377–384
Wang G, Park J, Sandhu R, Wang J, Gui X (2019) Dynamic trust evaluation model based on bidding and multi-attributes for social networks. Int J High Perform Comput Netw 13(4):436–454
Wu T, Wen S, Xiang Y, Zhou W (2018) Twitter spam detection: survey of new approaches and comparative study. Comput Secur 76:265–284
Zhang X, Ghorbani AA (2020) An overview of online fake news: Characterization, detection, and discussion. Info Process Manag 57(2):1–26
Zhang Z, Sun R, Zhao C, Wang J, Chang CK, Gupta BB (2017) Cyvod: a novel trinity multimedia social network scheme. Multimed Tools Appl 76(18):18513–18529
Zhang Z, Sun R, Wang X, Zhao C (2019a) A situational analytic method for user behavior pattern in multimedia social networks. IEEE Trans Big Data 5(4):520–528. https://doi.org/10.1109/TBDATA.2017.2657623
Zhang Z, Sun R, Choo K-KR, Fan K, Wu W, Zhang M, Zhao C (2019b) A novel social situation analytics-based recommendation algorithm for multimedia social networks. IEEE Access 7:117749–117760
Zheng R, Li J, Chen H, Huang Z (2006) A framework for authorship identification of online messages: writing-style features and classification techniques. J Am Soc Inf Sci Technol 57(3):378–393