Textual Analysis and Timely Detection of Suspended Social Media Accounts

Dominic Seyler, Shulong Tan, Dingcheng Li, Jingyuan Zhang, Ping Li
Cogntivie Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004, USA
{dominic.seyler, tanshulong2011, dingchengl, jyzhang.dut, pingli98}gmail.com

Abstract

Suspended[1] accounts are high-risk accounts that violate the rules of a social network. These accounts contain spam, offensive and explicit language, among others, and are incredibly variable in terms of textual content. In this work, we perform a detailed linguistic and statistical analysis into the textual information of suspended accounts and show how insights from our study significantly improve a deep-learning-based detection framework. Moreover, we investigate the utility of advanced topic modeling for the automatic creation of word lists that can discriminate suspended from regular accounts. Since early detection of these high-risk accounts is crucial, we evaluate multiple state-of-the-art classification models along the temporal dimension by measuring the minimum amount of textual signal needed to perform reliable predictions. Further, we show that the best performing models are able to detect suspended accounts earlier than the social media platform.

1 Introduction

Social media platforms are an important outlet for social interaction, but these platforms are riddled with deceitful users that engage in high-risk activities. These accounts are usually suspended by a platforms for various reasons: Twitter for instance, suspends accounts for spamming, security risks (e.g., account was compromised by a malicious entity) or user-reported, abusive behavior (Twitter 2021). Account suspension is a common and serious threat to the homogeneity of a platform, as it was shown that half of the Twitter accounts created in 2014 were ultimately suspended (Wei et al. 2015).

In order to maintain a better ecosystem, social network providers and scientific researchers have put great efforts into distinguishing regular from spam accounts (Almaatouq et al. 2016; Khan et al. 2018; Nilizadeh et al. 2017). However, in our manual analysis (Section 7) we find that only an estimated half of suspended accounts are spam accounts. Thus, spam classification methods seem to be inapplicable and specialized methods for suspended accounts are duly needed. Specialized detection methods, such as Volkova and Bell (2017), have proposed an array of features, but have not performed a thorough investigation of state-of-the-art deep models for the task and have not evaluated timely detection. Investigative work on suspended accounts, such as Thomas et al. (2011), have performed measurement studies related to the underground market of spamming, however they have not studied the textual content of suspended and benign accounts.

Many works in the social media domain, such as (Almaatouq et al. 2016; Karimi et al. 2018; Ruan et al. 2016), highlight the importance of behavioral characteristics, i.e., profile attributes and social interactions, to detect malicious users in Twitter. A major drawback is that behavioral properties are much harder to observe compared to the posting of text on social media. On top of that, most behavioral properties are not easily obtainable, since they require access to highly sensitive information about the users of a social network. Another challenge with behavioral features is the sparse integration of suspended accounts into the social network. It was shown that 89% of suspended accounts have fewer than ten followers on Twitter (Thomas et al. 2011). To avoid the effort of building a friendship network that is wide enough to be useful, malicious users usually make use of textual posts. Posts do not require the suspended account to integrate into the social network to reach a broad audience (Thomas et al. 2011) and do not necessarily exhibit abnormal behavioral patterns (Almaatouq et al. 2016). Furthermore, the malicious intend is much more difficult to obfuscate in textual output since the meaning of words cannot be changed at will. Therefore, we hypothesize that textual information of social media posts can play a significant role in helping to detect suspended accounts.

For the aforementioned reasons, we focus on the textual information on Twitter. We conduct statistical analyses to show that tweets themselves are helpful in distinguishing regular from suspended accounts (Section 3.2). For example, we find that suspended accounts are more likely to repeat words, which results in a much smaller vocabulary for the account (Figure 1b). However, we also see that in order to hide their intentions, malicious accounts behave similarly to regular users: They write posts with a similar amount of words as regular users (Figure 1d), retweet as often as normal users do (Figure 1g), etc. Moreover, we show that suspended accounts avoid using common spam words (Section 3.3). By exploiting a spam word list, we observe that

[1]Dominic Seyler is a PhD student in the Department of Computer Science at the University of Illinois – Urbana-Champaign. His work was conducted as a research intern at Baidu Research.

popular spam words cannot help differentiate suspended accounts from regular users (Figure 1i), even though many accounts are suspended for spreading spam (Thomas et al. 2011). To counteract, we automatically discover a latent word list (Table 3) that discriminates suspended accounts from regular users by employing an advanced topic model, namely Foreground/Background LDA (FB-LDA) (Tan et al. 2014). In our analysis we show that this list can help distinguish regular and suspended accounts significantly better than the list of common spam words (Figure 1j).

Num. Tweets	$\geqslant$25	$\geqslant$50	$\geqslant$75	$\geqslant$100
Perc. Accounts	87.6%	51.62%	37.39%	29.19%
Num. Accounts	147,209	86,750	62,834	49,060

Table 1: Number of tweets of suspended accounts.

We further study the problem of suspended account detection. Table 1 shows the statistics of suspended accounts we collected from Twitter. In our dataset, 87.6% of malicious accounts have posted more than 25 tweets before they get suspended by the platform, which can be seen as evidence that Twitter does not take immediate action when users start to tweet something deceitful. In this work, we aim to address this issue by detecting suspended accounts with as few tweets as possible. We propose to utilize different state-of-the-art learning models on the textual information of accounts. Our empirical experimental studies show that we can reliably distinguish (with an accuracy above 80%) suspended accounts from normal users, when only 10 tweets are observed at inference time. The experiments further show that we can reliably detect (with an accuracy of 78%) suspended accounts 15 tweets earlier than Twitter[2]. Even though we use Twitter as our data source, our methods are general enough to be applicable to any platform that uses textual content.

The contributions of this paper are the following:

1. We utilize the posting behavior of Twitter accounts to show that tweets themselves are helpful in distinguishing regular from suspended accounts. Given only the textual information of tweets, we study both the statistical and linguistic differences for suspended and regular accounts.
2. Based on the posting behavior of Twitter accounts, we study the task of detecting suspended accounts with as few tweets as possible. We implement several state-of-the-art deep learning models and compare them in terms of minimum textual information needed for making reliable predictions.
3. We study how early these methods can detect that an account will be suspended in the future. By using a sliding window over the posted tweets of an account, we investigate at what time point the predictive models can discover abnormal posts that indicate account suspension.
4. Given our analysis on posting behavior, we study how these insights can be incorporated into our models for suspended account detection. We do this by selecting the most promising statistical features and integrate them into the machine learning framework.

[2]Since suspended accounts on Twitter are partly user reported, this setup is not re-creating Twitter's automated detection algorithm. Instead, we see it as an extension to the existing automated detection framework, that implicitly incorporates user feedback.

2 Related Work

There are few works that have investigated suspended accounts. Similar to our goal, Volkova and Bell (2017) tackles the prediction of deleted and suspended accounts on Twitter. Even though the authors study the effectiveness of many features, the paper lacks an extensive study of state-of-the-art deep neural networks for the task. The work solely investigates LSTM networks, which we found are less suitable for suspended account detection than CNNs. The work treats suspended and deleted account detection as a prediction task, but does not evaluate any temporal aspects (e.g., how soon can these accounts be detected). Thomas et al. (2011) leverages suspended accounts to conduct an analysis of Twitter spam by investigating the activity and posting patterns of spammers, abuse of URL short-listing and how spam can be purchased in the underground economy. In contrast to our work, a linguistic analysis of suspended and benign accounts was not executed.

Substantial research has been undertaken into spam account detection: For instance, Sculley and Wachman (2007) performs an comparative analysis of different text classification methods for spam detection. Benevenuto et al. (2010) proposes to use numerical features, such as number of tweets, for classification. Lee, Caverlee, and Webb (2010) uses social honeypots to analyze spammers' behavior and to obtain classification labels. Martínez-Romo and Araujo (2013) detects spam messages by comparing two language distributions, estimated on the tweet and target webpage of a tweet's URL. Almaatouq et al. (2016) uses graph-based features, such as in/out-degree of the follower-followee graph, as features. Similarly, Khan et al. (2018) derives features based on authority and hub scores of users as features. Gupta et al. (2018) models users and tweet content in a heterogeneous network and uses an active learning framework to classify if new spam is part of these campaigns. Nilizadeh et al. (2017) clusters the tweet space by topics and derives features that measure the propagation of messages among different clusters. Adewole et al. (2019) proposes the use of profile-based features, such as location information and requires monitoring of users over extended periods. Sun et al. (2020) introduces a system for detecting Twitter spam in near real-time by combining features extracted from user accounts and tweet content. Even though spam detection is a related task to ours, we find in our manual analysis (Section 7) that only half of suspended accounts are spam accounts. Thus, spam classification methods are rendered inapplicable and specialized methods for suspended accounts are needed. Furthermore, we find that some previously proposed numerical features are not discriminative for distinguishing suspended accounts. Therefore, we propose and evaluate a number of novel features for this task.

Another line of work has focused on the detection of compromised accounts: Egele et al. (2017) creates behavioral

user profiles using certain user-specific characteristics, such as the time a user is active. VanDam, Tang, and Tan (2017) performs a measurement study and derives content-based features. Seyler, Li, and Zhai (2020) proposes a method that divides the tweet space randomly into compromised/benign tweets and uses the difference in language distributions as features. Karimi et al. (2018) utilizes LSTM networks to capture temporal dependencies to detect compromised accounts. VanDam et al. (2018) uses an unsupervised learning framework, where multiple views on a user profile (i.e., term, source, time and place) are encoded separately and then mapped into a joint space. This joint representation is then used to retrieve a ranking of compromised accounts. Building on this work, VanDam et al. (2019) uses and encoder-decoder framework to build low-dimensional feature vectors for users and tweets. The residual errors form both encoders are used in a supervised setting to predict compromised accounts. Here, residual errors will be higher if tweets were not written by a certain user, therefore indicating an account compromise. Ruan et al. (2016) detects behavioral anomalies by monitoring all internal and external interactions of Facebook accounts. Another work that makes use of behavioral profiles for is Velayudhan and Bhanu (2020), where frequency of benign and anomalous tweets are used for compromised account detection. Furthermore, Wang et al. (2020) models a user's expression habits by utilizing a supervised analytical hierarchy process for feature selection. A major drawback of these methods is that they rely on the integration of the accounts into the social network or use features that are only accessible by the social network provider.

Different works have analyzed malicious behavior on Twitter: Grier et al. (2010) presents the first comprehensive study into spam on Twitter. Thomas et al. (2014) studies the consequences if users fall victim to a malicious account takeover. Davidson et al. (2017) analyses and automatically identifies hate speech on Twitter by leveraging crowd sourcing and training a classifier on annotated data. In contrast, we study the linguistic and statistical properties of suspended accounts and provide novel insights for early detection, which were not covered by previous studies.

3 Analysis of Suspended Accounts

For our analysis we make use of a public Twitter dataset. First, we perform a statistical analysis on each account's posts and find that suspended account are distinguishable on select behavioral characteristics that have not been investigated by prior work. We then focus on the linguistic aspects of suspended accounts to see whether their use of language can discriminate them.

3.1 Dataset

Our textual information is drawn from Yang and Leskovec (2011), which is a large Twitter corpus of roughly 467 million posts from 20 million users, covering a seven month period from June to December 2009[3]. We derive the labels of whether an account is suspended directly from Twitter by checking if the URL of a user account is re-directed to https://twitter.com/account/suspended.

We perform a number of filter and sampling steps to the original dataset from Yang and Leskovec (2011): First, we remove all accounts with less than 20 tweets to ensure that the accounts have sufficient textual information for training and testing of our models. For the remaining accounts, we retrieve the user profile from Twitter by accessing each user's URL and then recording the HTTP response. From the server response we can also infer the exact date that the account was created (in case it was not suspended), which is embedded in the source code of each user's page.

We remove user accounts that were created after December 31, 2009 (the last date of the tweet crawl). Since we cannot infer the exact date of suspension for suspended accounts, we remove all suspended accounts that tweeted after December 1, 2009. Here, we hypothesize that suspended accounts that haven't tweeted for one month are more likely to be suspended before the end of our crawl. Although our method is language-independent, we chose to restrict the dataset to accounts that contain a majority of English tweets. Since only 5.1% percent of accounts are suspended, the distribution of positive and negative class labels is highly imbalanced. To counteract this, it is common in supervised learning frameworks to balance datasets to learn a better discriminative function. Balanced datasets were previously used for deleted and suspended account detection (Volkova and Bell 2017), compromised account detection (Karimi et al. 2018; VanDam et al. 2019; Seyler, Li, and Zhai 2020) and spam detection (Benevenuto et al. 2010; Lee, Caverlee, and Webb 2010; Nilizadeh et al. 2017; Adewole et al. 2019). To create a balanced dataset (i.e. the same amount of suspended and regular accounts), we perform undersampling where for each suspended account we select one regular account at random without replacement. The resulting dataset has a total of 166,642 accounts (see Table 4 for more details).

3.2 Statistical Analysis

As introduced earlier, some related work on studying suspended accounts has focused on social structure, information propagation or spam campaigns. Different from previous work, our analysis of suspended accounts focuses on a user's posting behavior. Posting behavior includes the time of posting and the posts' content, which is the most easily observable information to analyze or detect suspended accounts. Specifically, we first analyze various statistical characteristics of suspended accounts in our dataset ("†" feature is novel; "‡" feature is a distinguishing feature, which we integrate into our models).

Average Time Between Tweets†‡. Measures the time gap between postings (i.e., posting frequency). As found in previous work, 77% of accounts employed by spammers are

[3]Models trained on this dataset might not perform as effectively, when deployed as part of a current detection system. However, we expect that after re-training the models on more recent data the performance would be similar. The models we develop are also not specific to Twitter, so it's stricter character limit of 2009 is unlikely to affect performance.

suspended within one day (Thomas et al. 2011). So spammers are anxious to post in a short time window. To verify this, we plot the amount of users for different time gaps in Figure 1a. It can been seen that suspended accounts post much more frequently than regular users.

Vocabulary Size†‡. The vocabulary size for each account is shown in Figure 1b, where we find that suspended accounts have smaller vocabulary sizes. We hypothesize that malicious accounts often have narrow intentions and are forced to use a small set of words to deliver their message. In contrast, regular users may talk about more diverse topics, which we discuss in Section 3.3.

Average Number of URLs‡. Including URLs in posts is an effective way to post more information than what can be contained in a 140 character tweet. It was found that malicious actors need to use URLs to lure users to their target website, either for financial gains (Thomas et al. 2014) or spreading misinformation (Egele et al. 2017). Figure 1c confirms that suspended users use URLs more frequently.

Average Length of Tweets. Figure 1d shows the average character length of Tweets for each user. This measure seems not to be discriminative, since both kinds of accounts follow a similar distribution.

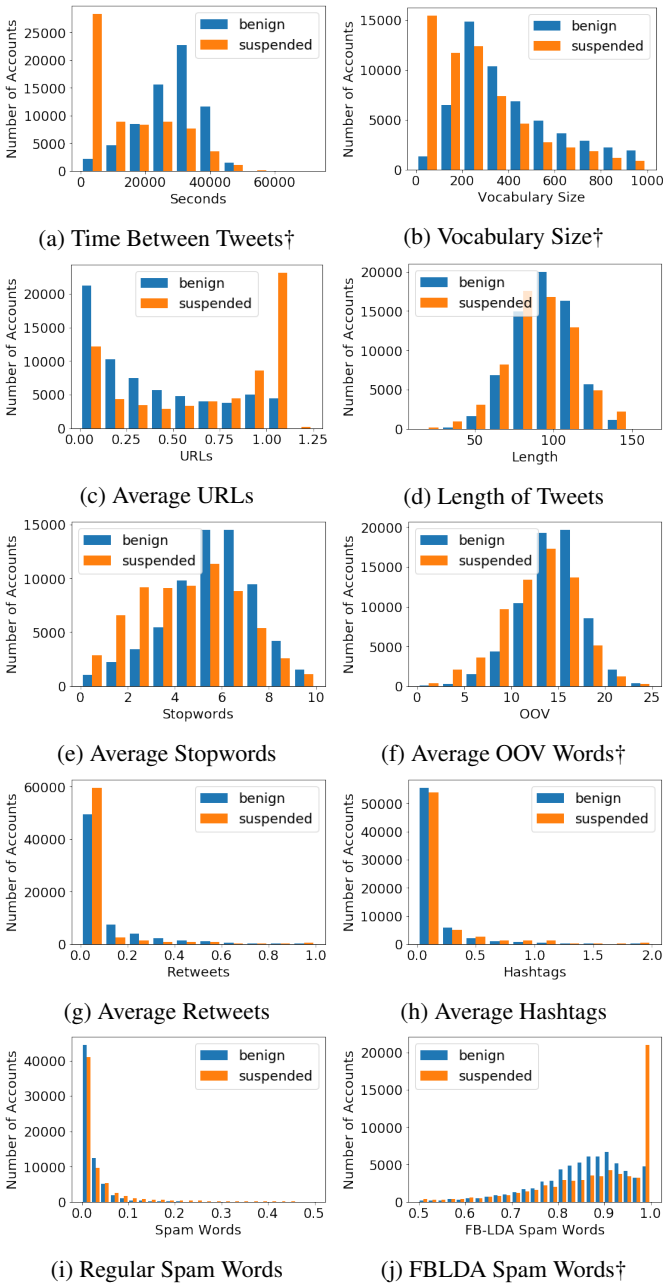
Average Number of Stopwords. We observe that suspended accounts use less stopwords (e.g., "and"), as shown in Figure 1e. We hypothesise that regular users are inclined to form complete sentences, whereas malicious accounts try to post as many informative words possible. However, the difference in distributions is much smaller when compared with the first three features above.

Average Number of OOV Words†. To find the number of out-of-vocabulary (OOV) words, we create a dictionary by removing the top 10% of words in terms of document frequency and by removing infrequent words (i.e. words that appear in less than 20 accounts). We then check the use of words outside our vocabulary for each user. The results are shown in Figure 1f. Surprisingly, suspended accounts have similar behaviors with regular accounts in OOV word usage.

Average Number of Retweets. In Figure 1g we explore the average number of retweets for each account. Suspended accounts have more retweets than regular accounts but the difference is not very distinguishable.

Average Number of Hashtags. Previous work found that malicious accounts tend to use popular hashtags of trending topics to attract a larger audience (Thomas et al. 2011). We count the average number of hashtags in tweets per user in Figure 1h. Even though hashtags seem to be effective for spreading tweets, suspended accounts do not use an increased amount of hashtags.

Percentage of Tweets that Contain Spam Words†‡. We check whether the suspended accounts are distinguishable from regular ones using common spam words. The spam word list we exploited is taken from Adewole et al. (2019). Tweets that contain at least one word in the spam list are counted for each account. The percentage of tweets that contain spam words for each account is shown in Figure 1i. Surprisingly, we can not distinguish suspended and not-suspended accounts from this perspective. Intuitively, suspended accounts would use more spam words because many engage in spamming behavior (Thomas et al. 2011). We hypothesise that spammers pretend to be regular users and avoid well-known spam words. Nonetheless, we believe that spammers leave some linguistic clues in textual output even when they avoid using certain words. In the following section, we mine a special vocabulary of suspended accounts with the help of an advanced topic model.



(a) Time Between Tweets†
(b) Vocabulary Size†
(c) Average URLs
(d) Length of Tweets
(e) Average Stopwords
(f) Average OOV Words†
(g) Average Retweets
(h) Average Hashtags
(i) Regular Spam Words
(j) FBLDA Spam Words†

Figure 1: Features related to tweets and content. Novel features are marked with a dagger (†).

Foreground Topics	Background Topics
celebrity gossip followers day using use fast music	news obama says new health times press care
dating free adult online years man seeking seeks	daily twittascope today online dress perfect prom evening
death lord stone guilty throws stoned blasphemer uttering	love day live playing listen today listening night
watch online free movie followers check site twitter	great check new today good day just thanks
money make twitter home online free people business	just legends nitto won branson race selling theatre

Table 2: Word with highest probability within topics using FB-LDA. Foreground topics are mined from suspended accounts, background topics are trained on regular accounts and can contain words from suspended accounts.

3.3 Linguistic Analysis

Our goal is to find out whether suspended accounts exhibit anomalies in their vocabulary and use of topics. For topic analysis we employ Foreground and Background LDA (FB-LDA) (Tan et al. 2014), which is a general framework for mining "special" topics in a corpus by contrasting a selected set of documents with a background set. The background set is believed to represent the general textual characteristics of the corpus. FB-LDA maintains two series of topic distributions, i.e., foreground topics and background topics. Each word from a background document samples a topic from the background topics. Each word from the contrasted document set has two choices: (a) choose a background topic; (b) choose a foreground topic if it is more frequent in the foreground but not common in the background set. The model inference produces two series of topic distributions. Usually, only foreground topics are considered for further use in downstream applications. In this paper, we consider all tweets of a user account as one document. Suspended accounts are input to FB-LDA as foreground documents and regular accounts are set as background documents. The number of topics is set to 20.

Table 2 shows the top words for selected foreground and background topics. Foreground topics (i.e., special topics from suspended accounts) talk about marketing, (illegal) downloads, adult dating and contain religious ranting. In contrast, background topics look more general and discuss daily life, news or politics. We also build a special vocabulary based on foreground topics from FB-LDA, which can be considered as an automatically generated spam word list. To build this vocabulary, we select the ten words with highest probability within all foreground topics. Using this new list we redraw Figure 1i in Figure 1j. It provides a very different result. Based on the special vocabulary mined by FB-LDA, suspended accounts are quite distinguishable from regular ones. We also compare ten words unique to each of the two lists in Table 3. The words obtained from FB-LDA show that suspended accounts spread their information by using many words that are not contained in common spam word lists.

FB-LDA		List from (Adewole et al. 2019)	
download	dating	save $	cash bonus
bucks	online	for free	click here
gossip	affair	insurance	gift certificate
xbox	cheating	great offer	winner
fantasy	software	opportunity	guarantee

Table 3: Words unique to each of the spam lists.

4 Suspended Account Detection Framework

Based on the statistic and linguistic analysis of the twitter dataset, we design a deep-learning framework for suspended account detection. Two main components are included in this framework, namely, word embeddings and deep nets. Word embeddings are generated from tweets given some pre-trained word vectors. Neural nets involve two alternatives, one from tweets and one combination of both tweets and selected auxiliary features described in Section 3.

4.1 Word Embedding

Words are represented as dense vectors in a low-dimensional vector space, where words with syntax and semantic relations tend to be close to each other. It is common to train a neural network on a large, unannotated text corpus to construct word embeddings. We use 200-dimensional Glove (Pennington, Socher, and Manning 2014) word vectors trained on Twitter to initialize the word embeddings. Compared to the n-gram (i.e., one-hot) method, features constructed from word embeddings are denser and less prone to overfitting on the training data.

4.2 Deep Neural Network Models

We implement a Convolutional Neural Network (CNN) model, which has been shown to be effective for text classification (Kim 2014). In the model, multiple convolution filters of different widths are sequentially applied to a sequence of word embeddings. Subsequently, max-pooling is applied to the output of the convolutional layer, the dropout operation is employed on the max-pooling output and the obtained result is fed into a fully-connected layer. Finally, a softmax layer is added. CNNs can reconstruct the high-level semantic features from input word embeddings and improve the training performance by weight sharing.

In our problem, we have a matrix of word vectors $\mathbf{X} \in \mathcal{R}^{N \times M}$ with filter weights $\mathbf{W} \in \mathcal{R}^{M \times O}$ and bias b, where N refers to the number of documents, M refers to the number of unique words in all tweets and O refers to the dimension of hidden layers. Then, a single convolution cell c_i is described as:

$$\mathbf{c}_i = f(\mathbf{W}^\top \mathbf{X} + b) \quad (1)$$

where f is a non-linear transformation function, such as ReLU (Dahl, Sainath, and Hinton 2013), which transforms

word vectors of M dimensions to $c_i \in \mathcal{R}^{O \times N}$ hidden vectors. The hidden vectors are stacked horizontally and the resulting set is denoted as:

$$\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_i, ..., \mathbf{c}_F] \tag{2}$$

where F refers to the number of filters and $\mathbf{C}$ refers to the collection of all convolutions. Namely, we conduct F such transformation via CNN and this will yield the feature matrix $C \in \mathcal{R}^{F \times O \times N}$. In order to reduce the complexity, max-pooling is employed to reduce C to a matrix $C_2 \in \mathcal{R}^{F \times O_2 \times N}$, where O_2 is smaller than O.

Features constructed by CNNs are undoubtedly powerful due to the multiple folds of filtering. However, the order of tweets are not taken into consideration. This may lead to some misrepresentation of the information. For example, a tweet may refer to a concept or person in a previous tweet. Therefore, we add a Long Short-Term Memory (LSTM) layer after the CNN layer, such that the output of the CNN layer (per tweet) is used as input to the LSTM. The last state of the LSTM is then used as a user representation of all her sequential tweets and fed into a fully-connected layer. Again, a softmax layer is added. with the following equations (for simplicity, we ignore the subscripts of features):

$$\begin{aligned}
\mathbf{i}_t &= relu(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{im}\mathbf{h}_{t-1}) + b_i \\
\mathbf{f}_t &= relu(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fm}\mathbf{h}_{t-1}) + b_f \\
\mathbf{o}_t &= relu(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{om}\mathbf{h}_{t-1}) + b_o \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \tanh(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{cm}\mathbf{h}_{t-1}) + b_c \\
\mathbf{h}_t &= \mathbf{o}_t \odot \mathbf{c}_t + \tanh(\mathbf{c}_{t-1})
\end{aligned} \tag{3}$$

Here t refers to each token of a tweet. As in other LSTM literature (Hochreiter and Schmidhuber 1997), $\mathbf{i}$ refers to input gates, $\mathbf{f}$ refers to forget gates, $\mathbf{o}$ refers to output gates, $\mathbf{c}$ still refers to convolution cell and yet, in LSTM, $\mathbf{c}$ is divided into T (the length of tweets) memory cells with each as $\mathbf{c}_t$, $\mathbf{h}$ refers to hidden states and b refers to bias. The forget gates $\mathbf{f}_t$ allow the model to selectively ignore past memory cell states and the input gates $\mathbf{i}_t$ allow the model to selectively ignore parts of the current input. The output gates $\mathbf{o}_t$ then allow the model to filter the current memory cell for its final hidden state. The combination of these gates present our model with the ability to learn long-term dependencies among features that were learned by CNN and reset these dependencies conditioned on certain inputs. Commonly, the LSTM transition function is an affine transformation followed by a point-wise nonlinearity like the hyperbolic tangent function.

4.3 Deep Neural Networks with Auxiliary Information Features

Besides word embeddings, we explore different auxiliary information features. In our previous study in Section 3.2 we found that most previously proposed features are not very discriminative in our problem setting. However, we do find a few promising novel features including average time between tweets, vocabulary size and an automatically generated list of spam words. To use both social media text and those features for making predictions, we make use of a feature concatenation mechanism to incorporate those features into the fully connected layer. Hence, with those features concatenated, our final predication layer becomes a layer of mixture features as follows:

$$\mathbf{p}_{t+1} = softmax\left(\mathbf{h}_T + \sum_{j=1}^{J} \mathbf{a}_j\right) \tag{4}$$

where T refers to the length of LSTM, $\mathbf{a}_j$ refers to the average values of auxiliary information features. Namely, besides $\mathbf{h}_T$, the fully connected layer sums up normalized average values of auxiliary features and together they make the final predictions.

5 Experiment Setup

This section discusses our experiment design and setup. We evaluate various deep neural network models on the test dataset in terms of classification accuracy and F_1 score. Additionally, we compare deep neural network models to "traditional" (i.e., SVM) models.

5.1 Experiment Dataset

For our experiments, we split the dataset introduced in Section 3.1 into training, development and testing, with 80%, 10% and 10% of the original size, respectively. The data are split by accounts rather than data size, meaning that no user accounts in one of the dataset will appear in another. This ensures that the information used in training will not be repeated in the evaluation datasets. An overview of the datasets is shown in Table 4 (we refer the reader to Section 3.1 for a discussion about why balancing of the data is necessary).

Dataset	Accounts	Suspended	Regular
Training	133,312	66,656	66,656
Development	16,664	8,332	8,332
Testing	16,666	8,333	8,333
Total	166,642	83,321	83,321

Table 4: The number of user accounts for training, development and testing datasets.

5.2 Classification Models

SVM: Similar to Sculley and Wachman (2007), we use n-grams as input to a linear support vector machine (SVM) for classification. The classifier's vocabulary is based on one- and two-grams from the training set. Inspired by Wang et al. (2017), we prevent over-fitting of our model by systematically reducing the size of the vocabulary. To eliminate stop words, we remove the top 10% of most frequent n-grams in the training data. From the remaining n-grams we select the 5,000 most frequent as the vocabulary. We try two variants of feature representations: In SVM^{TF}, each term is represented by its frequency in the document. In SVM^{TF*IDF}, each term is represented by its frequency multiplied by its inverse document frequency (Jones 2004), which is estimated using the training dataset.

CNN: We implement the model from Kim (2014). We apply filters with widths three, four and five with kernel size 100 to the concatenated tweet stream of a user.

LSTM: The LSTM network is directly applied at the word vector level. We compare uni-directional (LSTM) and bi-directional (LSTM^{bi}) models.

GRU: The GRU (Cho et al. 2014) is applied at the word vector level. For this model we also compare uni-directional (GRU) and bi-directional (GRU^{bi}) versions.

CNN-LSTM: Similar to Seyler et al. (2020), this model has a CNN layer applied on the tweet level and an LSTM applied to the CNN output (see Section 4 for details).

BERT: Taken from Devlin et al. (2019), BERT is a state-of-the-art language model that utilizes deep bi-directional transformers. We use the BERT$_{BASE}$ model and utilize the pre-trained model weights for uncased English text. Fine-tuning for our task is performed using the training portion of the dataset for four epochs. The model with the best performance on the development set is used for testing.

5.3 Model Implementation

We implemented all neural network models in the PaddlePaddle platform[4]. We used Scikit-learn 0.19.1 for the SVM models. We chose the rectified linear unit (ReLU) (Dahl, Sainath, and Hinton 2013): $relu(x) = \max(x, 0)$, as the activation function for all neural network nodes except the single neuron for prediction. For the prediction layer, we chose the sigmoid function: $sigmoid(z) = 1/(1 + e^{-z})$, which takes a real value input to an output in a range from 0 to 1. We selected Adam (Kingma and Ba 2015) as the kernel optimizer, binary cross-entropy as the loss function, batch size of 32 and dropout probability of 0.5. We use 200-dimensional Glove (Pennington, Socher, and Manning 2014) word vectors trained on Twitter to initialize the word embeddings. For BERT we use the implementation from Wolf et al. (2019).

5.4 Temporal Discretization

When designing our experiments, an important decision to make is how to represent and discretize time. We choose to use tweets as a proxy for time for the following reasons: Since tweets are posted sequentially in time, e.g. time(tweet$_i$) < time(tweet$_{i+1}$), we can still infer the relative temporal performance of our models. Thus, if model m_1 can detect a suspended account x tweets earlier than model m_2, we can infer that model m_1 is better suited for early detection. This can be done independently of the actual amount of time that has passed, which depends on the individual accounts posting frequency.

Another reason for using tweets as a proxy for time is that it gives fairer performance measurements, when time is discretized, or "binned". The problem originates in the underlying temporal distributions of the datapoints, which makes it impossible to find bins of equal size when time is discretized evenly. For example, if we decide to choose bins of one hour for our experiments, it will happen that some of the bins have more or less datapoints than others. This results in the performance measurements being distorted, as bins with only one correctly classified datapoint will show an accuracy of 100%, for instance.

We acknowledge that tweets are an imperfect representation of time. However, for the reasons mentioned above, we argue that temporal discretization based on tweets is still suitable for performance comparison of different classification models in our problem setting.

5.5 Minimize Textual Amount Exploited

In this experiment we train classification models by varying the amounts of tweets that are shown to the classifier at inference time. Our goal is to find the minimum amount of textual information needed to make reliable predictions. In a real-world setting, it is crucial to predict account suspension using as little information as possible, in order to contain the damage that a suspended account can cause. This experiment can also be seen as an investigation in the predictive power of our machine learning models, as more robust models make better predictions with fewer data.

In our experiments, we investigate the performance of our models from a minimum to a maximum amount of information. Since our application is within the realm of social media, we set our minimum and maximum to the two and 50 last posts of an account, respectively. We also show the immediate steps of size five and plot the performance in terms of prediction accuracy.

5.6 Early Detection

The experiment in the previous section explores the minimum amount of textual information needed. Even though this gives us a good way of estimating how quickly our classifiers can make predictions, it does not tell how early an account can be detected for suspension. It is obvious that a classifier which detects suspended accounts earlier is better suited for application in real-world settings. Bearing this in mind, we design the experiment to compare different classifiers and to give an estimate for how early reliable predictions can be made. We therefore choose to regard tweets immediately before account suspension as the "easiest" problem setting. The earlier in time the tweets were posted the harder the problem becomes. However, at the same time the classifier becomes more useful, since it can predict account suspension as a future event rather than a retrospective one.

To simulate this predictive task, we choose to employ a sliding window approach that moves backwards in time. Using a window of size W messages ensures that the model has enough data to base its prediction on. The task is made harder by sliding the window, starting from the time point of account suspension to an earlier time point. The sliding is done with a step size of s messages, applied i times. This way we simulate a classifier observing only W messages, $i * s$ messages before account suspension. If $i * s$ is larger than 0, the classifier predicts suspension occurring in the future.

[4]https://www.paddlepaddle.org.cn/

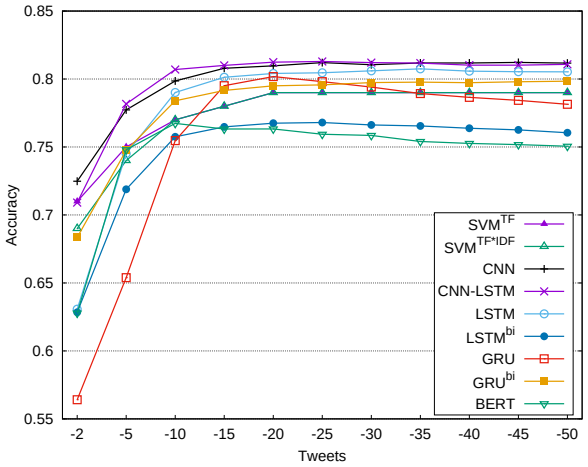


Figure 2: Performance of different classification models.

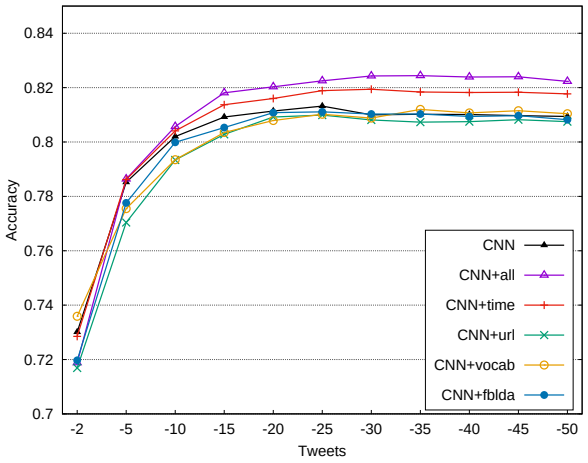


Figure 3: Performance of statistical features.

For our experiments, we set the window size W to 10, based on the results of the previous experiment in Section 5.5. The window is moved backwards up to the 50^{th} to last message. We show the performance in terms of prediction accuracy and F_1 score for sliding windows, with a step size s of five messages and $0 \leq i < 9$.

6 Experiment Results

We now evaluate different classification models along the dimensions of minimizing the exploited amount of textual information and early detection.

6.1 Comparison of Different Models

In Figure 2, we compare how different models perform when they are presented with varying amounts of information. We find that CNN, CNN-LSTM, SVM^{TF} and SVM^{TF*IDF} seem to perform most robustly, having good performance for only two tweets around 0.7 accuracy and increasing performance when adding up to 15 tweets. All five models experience few performance changes after 15 tweets, which leads us to believe that a window of 10-15 tweets is the sweet spot in terms of classification performance. It should be noted that although performance hovers around 0.7 when the models are only presented with two tweets, it shows that even small amounts of text give strong signals to the model. CNN and CNN-LSTM outperform the SVM-based models for all data points, reaching a maximum performance of 0.8122 and 0.8129.

Interestingly, the LSTM and GRU models perform subpar for small amounts of tweets. This might be due to the fact that RNN based models perform better for longer sequences of text. This reasoning would also explain their strong increase once more text is shown to the model. Surprisingly, the BERT model can only outperform the weak GRU and LSTM^{bi} models, when less than 10 tweets are observed. The performance peaks when about 10 tweets are observed but then slowly decreases. We hypothesize that the restriction of input tokens in the BERT model prevents it from benefiting from the additionally observed tweets. Another reason for the low performance might be the frequency of misspelled words in social media text. Even though BERT uses subword embeddings for out-of-vocabulary tokens, it might not be as effective as our word embeddings that have been pre-trained on in-domain (i.e., Twitter) text. We further find that "simple" textual signals, as captured by SVM and CNN, seem to be sufficient for this classification task. Thus, the additional expressiveness of the language model that is introduced by BERT does not seem helpful. For the aforementioned reasons, we will focus on CNN and CNN-LSTM in our subsequent experiments.

When comparing the SVM baselines, there seems to be almost no difference in performance. However, SVM^{TF} performs slightly better for very small amounts of data, since it outperforms SVM^{TF*IDF} for data points -2 and -5. Following this observation, we evaluate SVM^{TF} in the subsequent experiments.

6.2 Features from Statistical Analysis

From Section 3, we select the most promising features, which are: (1) average time between tweets (*time*) (2) average number of URLs (*url*) (3) vocabulary size (*vocab*) and (4) percentage of tweets that contain words obtained with FB-LDA (*fblda*). In Figure 3, we test the individual impact of each feature, as well as concatenating all features (*all*) when added to the CNN model. We find *time* to be the strongest feature since it performs equally or better than the standard CNN model. Features *url*, *vocab* and *fblda* do not increase the performance when added individually (this might be because CNN related models have taken advantage of this information already). However, when added to *time* the performance can still be improved, especially when more than ten tweets are used as input. Using all features the model outperforms CNN with up to 1.43 percentage points.

6.3 Early Detection

The purpose of this experiment is to show how early we can detect a future account suspension. As discussed in Section 5.4, we measure time in terms of messages posted by

Window	SVM			CNN			CNN+all			CNN-LSTM			CNN-LSTM+all		
	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R
[-10, 0]	.77	.79	.74	.7907	.82	.76	.7990**	.83	.77	.8070	.81	.80	.8023	.83	.77
[-15, -5]	.75	.76	.75	.7784	.79	.76	.7923**	.80	.78	.7892	.76	.81	.7968**	.80	.78
[-20, -10]	.75	.75	.75	.7680	.77	.76	.7842**	.77	.79	.7762	.74	.81	.7810**	.78	.78
[-25, -15]	.74	.73	.75	.7592	.75	.77	.7785**	.76	.80	.7677	.72	.82	.7805**	.77	.79
[-30, -20]	.73	.71	.75	.7473	.72	.77	.7672**	.73	.80	.7551	.69	.82	.7693**	.73	.81
[-35, -25]	.73	.67	.80	.7397	.68	.81	.7572**	.69	.84	.7437	.66	.85	.7575**	.68	.84
[-40, -30]	.72	.64	.83	.7320	.65	.84	.7424**	.65	.87	.7354	.63	.87	.7421*	.65	.87
[-45, -35]	.72	.62	.85	.7206	.62	.85	.7313**	.62	.88	.7267	.61	.89	.7345**	.63	.88
[-50, -40]	.71	.61	.87	.7171	.61	.87	.7276**	.61	.90	.7213	.60	.90	.7289	.61	.90

Table 5: Results for early detection. Statistical features significantly improve performance according to McNemar test with p-value $\leq \mathbf{0.05}$ (*) and p-value $\leq \mathbf{0.01}$ (**).

how to post rss feeds to twitter ? <link> alternate to adwords campaigns try using <link> top 10 ghetto weddings ...
debt settlement advice - where to get free advice and find the <unk> can a personal loan help you with your finances ? ...
subscribe and find free teens <link> #admitit subscribe for free girls <link> #egibitow subscribe and find free teens...
rapid fat loss handbook <unk> meryl streep struggles to lose movie weight and tips that boost ...
atx computer cases spy equipment cell phones . visit us today and save <link> . home security cameras security camera systems ...
yupp u can watchh g.i. joe : the rise of cobra moviie online <link> #welovethenhs just watchhed the time traveler's wife moviie ...
i rated a youtube video (5 out of 5 stars) - - techno mix 2009 <unk> i favorited a youtube video ...
how domino "s", <unk> measures social media success : " are my sales up ? am i making money ? am i having fun ? that "s", it ...

Table 6: A sample of false negative user accounts that were correctly identified by our method.

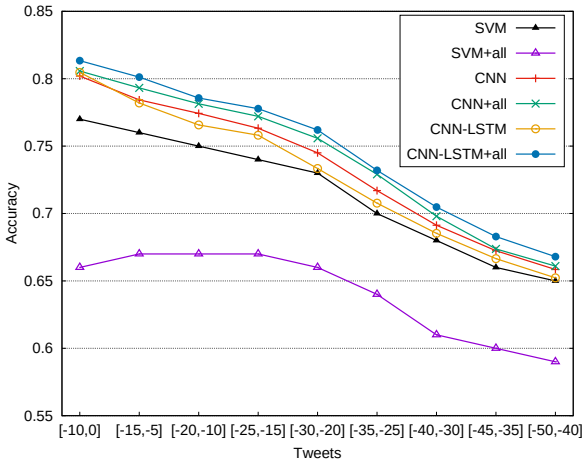


Figure 4: Best performing models on early detection.

an account. In Figure 4, we compare the best performing models from Section 6.1 and Section 6.2 for this task. The most recent time point is at the origin of the plot, whereas the furthest away time point is on the end of the x-axis.

All models perform worse the further we step back in time, meaning that the task becomes more and more difficult. Again, we find that the SVM baselines are outperformed by the deep-learning models. Surprisingly, adding the statistical features we introduced in Section 3 to the SVM model results in a large performance decrease. However, the statistical features help both CNN and CNN-LSTM models. The most significant gains are made when these features are added to the CNN-LSTM model, which performs best overall in this task. We find that predictions with this model are quite reliable even up to 30 tweets in the past, where the performance is still above 0.75 in terms of accuracy. The accuracy ranges from 0.8134 to 0.668 for this model. The CNN-based model performs slightly worse from 0.8057 to 0.661.

Table 5 lists F_1, precision (P) and recall (R) scores of our best performing models (we omitted SVM+all due to inferior performance to SVM). For F_1 scores, we observe a similar pattern as for classification accuracy. The deep learning models outperform SVM for all data points, but the difference becomes smaller the further we move into the past. CNN-LSTM+all also seems to perform superior in terms of F_1 scores. Adding the statistical features to the CNN model increases performance of each data point by an average of 1.4 percentage points. The biggest increases (close to 2 percentage points) can be seen between windows [-25,-15] and [-30,-20]. Adding statistical features significantly improves performance for all time points in the CNN model (p-value ≤ 0.01 by McNemar test (Dietterich 1998)). Similarly, adding statistical features to the CNN-LSTM model significantly improves performance for time points within windows [-15,-5] and [-45,-35]. We take this as evidence that our features are especially effective on earlier data points and hence suitable for early detection.

While further evaluating Precision and Recall in Table 5, we find that both metrics improve in the CNN-based models, compared to SVM. Similar to our earlier observation, the improvement is higher for windows less far in the past. Interestingly, Precision suffers when the windows move further into the past, whereas Recall increases. Adding the statistical features improves Precision and Recall for CNN. For CNN+LSTM, it seems to improve Precision at the cost of a minor reduction in Recall.

7 Error Analysis

We perform a manual error analysis on false negatives and show concrete examples that were correctly identified by our method. We find that there are a number of false negatives in the dataset that our classifier correctly identified as positive. In Table 6 we give the output of eight example accounts that we manually identified as false negatives among a set of 100 randomly selected not-suspended accounts. In our manual investigation we find seven more of such accounts, which leads us to estimate that roughly 15% of accounts found by our classifier are wrongly labeled as benign in the gold standard and should be suspended by Twitter.

Upon manual investigation, we find that the topics and motives of these accounts are quite variable but the majority of them could be categorized as spam. We also find that in some accounts, which our classifier labels as suspended, users that talk very vulgarly and derogatory. It is surprising that these accounts were not caught by the social media platform, since a simple list of curse words would have identified them. To summarize, we regard these findings as preliminary evidence that a non-insignificant amount of accounts in our dataset are false negatives (i.e. mislabeled) in the gold standard (i.e. ground truth from Twitter).

We further categorize 100 random accounts that are labeled as suspended by the ground truth. Manual evaluation is necessary, since the reason for account suspension is not known from the label. We chose the following categories and annotation guidelines:

- **Spam**: promotions, download links, etc.
- **Offensive Language**: crude language and insults
- **News**: current events, weather forecasts, etc.
- **Adult**: suggestive language, e.g., "xxx"
- **Spiritual**: religious or preachy language
- **Benign**: regular user account
- **Unsure**: no obvious reason, or mix of categories

	Spam	**Offen.**	**News**	**Adult**	**Spirit.**	**Ben.**	**Uns.**
TP	45	6	4	6	6	6	6
FP	3	2	1	0	2	9	4
Ratio	48%	8%	5%	6%	8%	15%	10%
FDR	.06	.25	.20	0	.25	.60	.40

Table 7: Categories of suspended accounts.

Table 7 shows the results. Since we only investigate accounts labeled as suspended, we show true positives (TP), false positives (FP), percentage of accounts of a certain category (Ratio) and false discovery rate (FDR), which is defined as $\frac{FP}{FP+TP}$. From the table we find that almost half of the accounts are spam accounts, which our classifier identifies robustly, with an FDR of 0.06. Offensive, spiritual and news account make up a combined 21% of accounts and are classified with 0.20 to 0.25 FDR. No misclassification happens for the adult category. We speculate that these accounts can be found easily, due to their very distinct vocabulary (e.g., "xxx"). The classes with the highest FDR are benign and unsure, with 0.6 and 0.4 FDR, respectively. We argue, that the higher errors in these accounts are due to the general difficulty of distinguishing them, which is a hard task for humans, as well.

8 Conclusion and Future Work

We performed a linguistic and statistic analysis into suspended social media accounts and showed that suspended accounts differ from regular accounts in behavioral and content-related aspects when they post textual messages. Our linguistic study highlighted the manifold textual differences of suspended accounts, which often discuss topics that are measurably different from benign accounts. Using advanced topic modeling, we were able to automatically derive a suspended account spam word list, which we showed to be much better at distinguishing suspended from benign accounts compared to existing word lists targeted exclusively at detecting spam. From our statistical study, we derive four features and showed how they significantly improve a deep-learning-based suspended account prediction framework. Our method requires only a small amount of textual information (about ten short sentences/tweets) along with their timestamps as input. In our experiments, we showed that this signal is sufficient to perform reliable predictions (with over 80% accuracy in some cases) and that our model is able to detect suspended accounts earlier than the social media platform. In our manual error analysis, we find that our classifier performs robustly over various categories of suspended accounts.

In what follows, we present a non-exhaustive list of future directions enabled by this work:

- **Real-word deployment:** In one potential application, our method would be used to generate alarms about possible account suspension in the future, so that humans can examine them. This system could be deployed on both "sides" of social media: (1) the platform provider (e.g., Twitter) could incorporate our models to inform its content moderators of high-risk accounts and (2) users of a social media platform could be warned before posting critical content. Even though there may be inevitable false alarms, our method exhibits sufficient accuracy to be practically useful.
- **Further study the automatic creation of spam word dictionaries:** In this work we have shown that FB-LDA can be leveraged for creating spam word lists. In the future, we can envision a more principled study into hyperparameters, such as, number of topics and optimal cut-off points for number of words within a topic. It would also be interesting to see how efficient the generated spam word list would perform on a spam detection task. Therefore, an evaluation on spam detection datasets would be desirable.
- **Incorporation of global lexical information:** Our current methods are limited in the way they only consider local lexical information (besides from the spam word dictionary). Statistical and textual features are derived directly from a user account and do not consider the "greater picture", meaning the entire document collection. In this way, our models potentially miss patters that are evident across multiple accounts but are hidden when accounts

are examined in isolation. We therefore assume that there is much potential in extending the existing work using global features, for instance, using topic modeling.

- **Incorporate features based on social graph:** Related to the previous observation where we pointed out that accounts are classified locally, we can imagine incorporating follower-followee relationships into the model. These signals based on the social graph should be somewhat orthogonal to our text-based features and their incorporation could lead to potentially further performance improvements.
- **Domain Adaptation:** Another area for improvement is the domain adaption of our classification models. In this work, "out-of-the-box" word embeddings trained on general text corpora are used, but they can be less effective when applied to domain-specific settings. Incorporating a method such as domain adaptation through backpropagation (Ganin and Lempitsky 2015), or domain adaptation of word embeddings (Seyler and Zhai 2020), could further benefit our existing classification framework.
- **Understand evasion tactics:** In the literature, there currently exists only limited understanding of how text-based detection techniques of social media abuse can be evaded. As bad actors are constantly modifying their behavior to fool detection systems, it would be crucial to further study their malicious actions over time and test the robustness of our text-based features in an ever-changing environment.

References

Adewole, K. S.; Anuar, N. B.; Kamsin, A.; and Sangaiah, A. K. 2019. SMSAD: a framework for spam message and spam account detection. *Multim. Tools Appl.* 78(4): 3925–3960.

Almaatouq, A.; Shmueli, E.; Nouh, M.; Alabdulkareem, A.; Singh, V. K.; Alsaleh, M.; Alarifi, A.; Alfaris, A.; and Pentland, A. S. 2016. If it looks like a spammer and behaves like a spammer, it must be a spammer: analysis and detection of microblogging spam accounts. *Int. J. Inf. Sec.* 15(5): 475–491.

Benevenuto, F.; Magno, G.; Rodrigues, T.; and Almeida, V. 2010. Detecting spammers on twitter. In *Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS)*, 12–22.

Cho, K.; van Merrienboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST@EMNLP)*, 103–111. Doha, Qatar.

Dahl, G. E.; Sainath, T. N.; and Hinton, G. E. 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8609–8613. Vancouver, Canada.

Davidson, T.; Warmsley, D.; Macy, M. W.; and Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the Eleventh International Conference on Web and Social Media (ICWSM)*, 512–515. Montréal, Canada.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–4186. Minneapolis, MN.

Dietterich, T. G. 1998. Approximate Statistical Tests For Comparing Supervised Classification Learning Algorithms. *Neural Comput.* 10(7): 1895–1923.

Egele, M.; Stringhini, G.; Kruegel, C.; and Vigna, G. 2017. Towards Detecting Compromised Accounts on Social Networks. *IEEE Trans. Dependable Secur. Comput.* 14(4): 447–460.

Ganin, Y.; and Lempitsky, V. S. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 1180–1189. Lille, France.

Grier, C.; Thomas, K.; Paxson, V.; and Zhang, C. M. 2010. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*, 27–37. Chicago, IL.

Gupta, S.; Khattar, A.; Gogia, A.; Kumaraguru, P.; and Chakraborty, T. 2018. Collective Classification of Spam Campaigners on Twitter: A Hierarchical Meta-Path Based Approach. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web (WWW)*, 529–538. Lyon, France.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.* 9(8): 1735–1780.

Jones, K. S. 2004. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation* 60(5): 493–502.

Karimi, H.; VanDam, C.; Ye, L.; and Tang, J. 2018. End-to-End Compromised Account Detection. In *Proceedings of the IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 314–321. Barcelona, Spain.

Khan, M. U. S.; Ali, M.; Abbas, A.; Khan, S. U.; and Zomaya, A. Y. 2018. Segregating Spammers and Unsolicited Bloggers from Genuine Experts on Twitter. *IEEE Trans. Dependable Secur. Comput.* 15(4): 551–560.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Doha, Qatar.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA.

Lee, K.; Caverlee, J.; and Webb, S. 2010. Uncovering social spammers: social honeypots + machine learning. In *Proceeding of the 33rd International ACM SIGIR Conference*

on Research and Development in Information Retrieval (SIGIR), 435–442. Geneva, Switzerland.

Martínez-Romo, J.; and Araujo, L. 2013. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Syst. Appl.* 40(8): 2992–3000.

Nilizadeh, S.; Labreche, F.; Sedighian, A.; Zand, A.; Fernandez, J. M.; Kruegel, C.; Stringhini, G.; and Vigna, G. 2017. POISED: Spotting Twitter Spam Off the Beaten Paths. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 1159–1174. Dallas, TX.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Doha, Qatar.

Ruan, X.; Wu, Z.; Wang, H.; and Jajodia, S. 2016. Profiling Online Social Behaviors for Compromised Account Detection. *IEEE Trans. Inf. Forensics Secur.* 11(1): 176–187.

Sculley, D.; and Wachman, G. 2007. Relaxed online SVMs for spam filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 415–422. Amsterdam, The Netherlands.

Seyler, D.; Li, L.; and Zhai, C. 2020. Semantic Text Analysis for Detection of Compromised Accounts on Social Networks. In *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 417–424. Virtual Event.

Seyler, D.; Shen, J.; Xiao, J.; Wang, Y.; and Zhai, C. 2020. Leveraging Personalized Sentiment Lexicons for Sentiment Analysis. In *Proceedings of the 2020 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR)*, 109–112. Virtual Event, Norway.

Seyler, D.; and Zhai, C. 2020. A Study of Methods for the Generation of Domain-Aware Word Embeddings. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*, 1609–1612. Virtual Event, China.

Sun, N.; Lin, G.; Qiu, J.; and Rimba, P. 2020. Near Real-Time Twitter Spam Detection with Machine Learning Techniques. *International Journal of Computers and Applications* 1–11.

Tan, S.; Li, Y.; Sun, H.; Guan, Z.; Yan, X.; Bu, J.; Chen, C.; and He, X. 2014. Interpreting the Public Sentiment Variations on Twitter. *IEEE Trans. Knowl. Data Eng.* 26(5): 1158–1170.

Thomas, K.; Grier, C.; Song, D.; and Paxson, V. 2011. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference (IMC)*, 243–258. Berlin, Germany.

Thomas, K.; Li, F.; Grier, C.; and Paxson, V. 2014. Consequences of Connectivity: Characterizing Account Hijacking on Twitter. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 489–500. Scottsdale, AZ.

Twitter. 2021. About suspended accounts. https://help.twitter.com/en/managing-your-account/suspended-twitter-accounts. Accessed: 2021-04-12.

VanDam, C.; Masrour, F.; Tan, P.; and Wilson, T. 2019. You have been CAUTE!: early detection of compromised accounts on social media. In *Proceedings of International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 25–32. Vancouver, Canada.

VanDam, C.; Tan, P.; Tang, J.; and Karimi, H. 2018. CADET: A Multi-View Learning Framework for Compromised Account Detection on Twitter. In *Proceedings of the IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 471–478. Barcelona, Spain.

VanDam, C.; Tang, J.; and Tan, P. 2017. Understanding compromised accounts on Twitter. In *Proceedings of the International Conference on Web Intelligence (WI)*, 737–744. Leipzig, Germany.

Velayudhan, S. P.; and Bhanu, S. M. S. 2020. UbCadet: detection of compromised accounts in twitter based on user behavioural profiling. *Multim. Tools Appl.* 79(27-28): 19349–19385.

Volkova, S.; and Bell, E. 2017. Identifying Effective Signals to Predict Deleted and Suspended Accounts on Twitter Across Languages. In *Proceedings of the Eleventh International Conference on Web and Social Media (ICWSM)*, 290–298. Montréal, Canada.

Wang, X.; Tang, H.; Zheng, K.; and Tao, Y. 2020. Detection of compromised accounts for online social networks based on a supervised analytical hierarchy process. *IET Inf. Secur.* 14(4): 401–409.

Wang, Y.; Seyler, D.; Santu, S. K. K.; and Zhai, C. 2017. A Study of Feature Construction for Text-based Forecasting of Time Series Variables. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, 2347–2350. Singapore.

Wei, W.; Joseph, K.; Liu, H.; and Carley, K. M. 2015. The fragility of Twitter social networks against suspended users. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 9–16. Paris, France.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* .

Yang, J.; and Leskovec, J. 2011. Patterns of temporal variation in online media. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining (WSDM)*, 177–186. Hong Kong, China.