applied sciences

MDPI

Article

A Deep Neural Network Technique for Detecting Real-Time Drifted Twitter Spam

Amira Abdelwahab [1,2,*] and Mohamed Mostafa [2]

[1] Department of Information Systems, College of Computer Science and Information Technology (CCSIT), King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia
[2] Department of Information Systems, Faculty of Computers and Information, Menoufia University, Shibin Al Kawm 32511, Egypt; mohamed.saad@ci.menofia.edu.eg
* Correspondence: a.ahmed@kfu.edu.sa

Abstract: The social network is considered a part of most user's lives as it contains more than a billion users, which makes it a source for spammers to spread their harmful activities. Most of the recent research focuses on detecting spammers using statistical features. However, such statistical features are changed over time, and spammers can defeat all detection systems by changing their behavior and using text paraphrasing. Therefore, we propose a novel technique for spam detection using deep neural network. We combine the tweet level detection with statistical feature detection and group their results over meta-classifier to build a robust technique. Moreover, we embed our technique with initial text paraphrasing for each detected tweet spam. We train our model using different datasets: random, continuous, balanced, and imbalanced. The obtained experimental results showed that our model has promising results in terms of accuracy, precision, and time, which make it applicable to be used in social networks.

Keywords: spam detection; deep learning; semantic similarity; social network security

Citation: Abdelwahab, A.; Mostafa, M. A Deep Neural Network Technique for Detecting Real-Time Drifted Twitter Spam. *Appl. Sci.* **2022**, *12*, 6407. https://doi.org/10.3390/app12136407

Academic Editors: Dionisis Margaris and Stefanos Ougiaroglou

Received: 15 May 2022
Accepted: 20 June 2022
Published: 23 June 2022

1. Introduction

Currently, many internet users can impart information and work together inside online social networks (OSNs). However, Twitter is viewed as the most well-known informal community which offers free blogging services for clients to publish their news and thoughts inside 280 characters. Clients can follow others through various platforms [1]. Consistently, a huge number of Twitter clients share their status and news about their disclosures [2]. Moreover, the Twitter platform additionally attracts criminal records (spammers) that can tweet spam substances, which may incorporate destructive URLs. This could divert clients to malevolent or phishing sites for bringing in cash misguidedly [3,4] by assaulting the client's profile. As Twitter set caps for the length of the characters of tweets, this makes spammer swindle clients by putting cheat content or malicious URL to divert them for the outside site [5]. In an investigation studying the correlation between both email and social spam, the click-through rate of Twitter spam was found to reach 0.13%, in spite of the fact that email spam arrives at 0.0003–0.0006% [6]. Moreover, social spam is viewed as increasingly perilous and cheats a lot of clients [7].

To tackle this problem, many researchers are focusing on detecting spammers by discovering the statistical features of spammers on both messaging and account levels. These messaging detection approaches focus on checking tweet content to find keyword patterns, hashtags, and URLs. These approaches are shown to be effective, but real-time detection is needed to solve the huge number of messages which are posted per hour. The account level approaches focus on extracting statistics and info about the behavior of each account to classify whether they are spam accounts or legitimate users. However, an experimental study was conducted to examine whether the statistical features changed over time. The experimental results proved that the statistical features are changed over time.

Most of the researchers are focusing on collecting these features and trying to experience spammers priorities, ignoring that these features drift over time. However, spammers will try to tackle all these features. In this paper, an effective technique has been proposed to tackle the aforementioned limitations. Our proposed technique focuses on the content of each tweet in addition to the statistical features. Moreover, it has an auto-learning capability to find the features which make it able to classify each tweet as spam or not with high accuracy in a reasonable time.

Accordingly, these challenges inspire us to investigate this problem to contribute to spam detection approaches. To cope with this problem, we maintain a framework that contains three stages to detect spammers:

- Fast filter mode classifier to determine whether each input tweet is spam or not.
- Every filtered spam tweet is paraphrased to generate a new spam sentence with different definition with the same meaning.
- Ensemble deep learning methods are collected in addition to the statistical features to decide the output of the classifier.

The rest of this manuscript is organized as follows. Section 2 briefly discusses the literature review on Twitter spam detection. Section 3 clarifies the problem statement of spam drift in detail. Section 4 explains our proposed detection framework. Section 5 discusses our experiments and results. Finally, conclusions are represented in Section 6.

2. Literature Review

Many studies have been performed to improve spam detection challenges. These studies can be organized into three categories [8], syntax analysis, feature analysis, and blacklist techniques, as shown in Figure 1.

Most of the research applied blacklisting techniques based on URLs in the tweets using any third-party tools, such as Trend Micro or Google safe browsing. However, S. Savage [9] creates a lightweight technique for spam detection, while [10] filtered tweets based on checking URLs in tweets, username patterns, and hashtags.

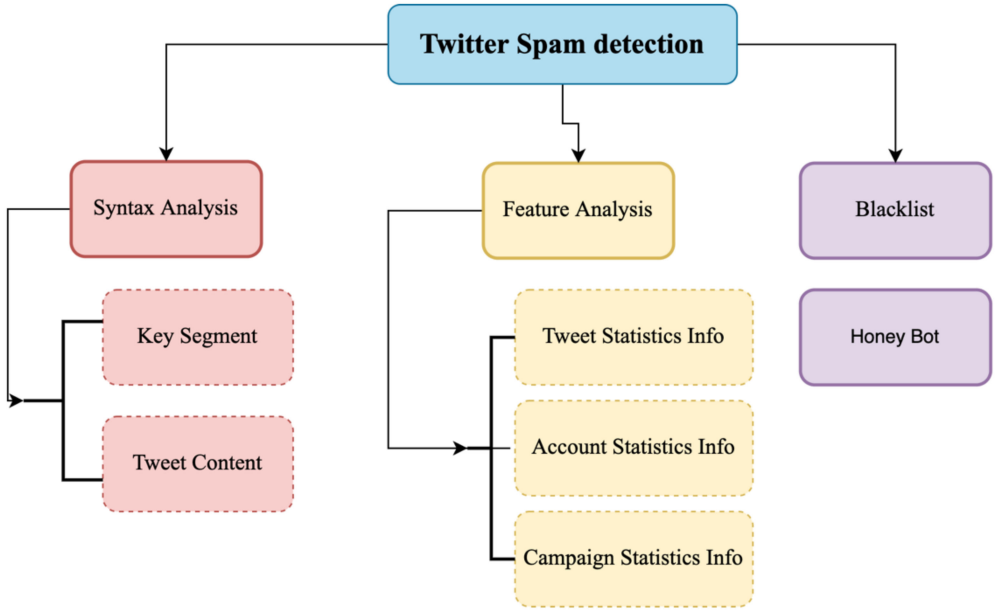

Figure 1. Twitter spam detection taxonomy.

Consequently, a lot of researchers have applied machine learning (ML) techniques in their works [11–14] and extracted some features of users, such as number of followings, username pattern, and account creation in addition to features of content, such as length of tweets, number of hashtags, and hashtags pattern. Authors in [11] employed honey pots

to collect spammers' profiles to extract statistical features using different ML algorithms, such as Decorate and Random some space. However, Benevento et al. attempted to detect spammers by using a support vector machine (SVM) algorithm [12]. These features can be easily fabricated as spammers can buy fake spammers' followers and followings. Thus, some studies [15] depend on a social graph to tackle the problem of fabrication by calculating distance and connectivity of each tweet between sender and receiver to examine whether it is spam. Yang et al. [16] built a more robust feature using a bidirectional link ratio between centrality and local cluster coefficient with performance 99% true positive, while [17] provides a new solution that can detect most campaigns and classify each of them into spam or not spam using deep learning techniques and semantic similarity methods.

Most of the described methods focus on detecting spam tweets based on some statistical features. Some studies employ syntax analysis, while a spam dataset based on hashtags was created by [18], in which authors collected 14 million tweets and classified them using five different techniques. Sedhi and son [19] utilized a package of four lightweight techniques to detect spam at tweet level using part of speech tag, content-based, sentiment, and user-based features, using a word vector as the universal feature of their task. Le and Mikulov [20] have deployed a deep learning method by constructing a tweet vector by combining the word vector with the document vector to classify the neural network.

In [21], the authors employ the horse herd optimization algorithm (HOA), inspired by nature optimization algorithms. This algorithm emulates the social exhibitions of horses at various ages. The idea behind this study has a great performance result on complex problems, specifically with high dimensions, solving many dimension problems with low cost based on time, performance, and complexity (up to 10,000 dimensions). The researcher attempts to find the best solution by employing the multiobjective opposition-based binary which gave good results compared with similar approaches. However, it still depends on statistical functions which can deviate over time as explained.

The study by Abayomi-Alli [22] used the ensemble approach to detect SMS spam. This approach depends on two pipeline the BI-LSTM (Bidirectional Long-Short Term Memory) network which produce accurate results in text classification tasks and the classical machine learning methods. However, this approach does not employ any attention mechanism in the BI-LSTM network, which causes this approach to suffer in long sentences of more than 8 words.

Many different extraction methods have been used for representing tweets, such as [23]. In this reference, authors analyzed people's sentiments collected through tweets. They employed three different feature extraction methods, domain-agnostic, fastText-based, and domain-specific, for tweet representation. Then, an ensemble approach was proposed for sentiment analysis by employing three CNN models and traditional ML models, such as random forest (RF), and SVM using the Nepali Twitter sentiment dataset, called NepCOV19Tweets. Their models achieve 72.1% accuracy by employing a smaller feature size (300-D). However, these models have two limitations. First, they are complex and need high computational resources for implementation. Second, their methods are based on only semantic features.

In addition, authors in [24] analyzed people's sentiments using three feature extractions, term frequency-inverse document f(TF-IDF), fastText, and a combination of these two methods as hybrid features for representing COVID-19 tweets. Then, they validated their methods against different ML techniques. Their SVM model obtained the highest accuracy on both TF-IDF (65.1%) and hybrid features (72.1%). The major limitation of this model is its high computational complexity.

TF-IDF [25] may be used to vectorize text into a format that is more suitable for machine learning and natural language processing approaches. It is a statistical measure that we can apply to terms in a text and then use to generate a vector, whereas other methods, such as word2vec [26], will provide a vector for a term and then extra effort may be required to transform that group of vectors into a single vector or other format. Another approach is Bidirectional Encoder Representations from Transformers (BERT),

which converts phrases, words, and other objects into vectors using a transformer-based ML model [27]. However, BERT's design also includes deep neural networks, which means it can be significantly more computationally expensive than TF-IDF.

Because our proposed framework will be used with highly intensive data applications, we had to choose a high-performance and quick feature extraction method. TF-IDF produces high accuracy relative to our framework, so we decided to build our model with it.

Most of the mentioned studies focus on extracting the features that can help them find the spammers, but they ignore a very important problem, which is "spam drift", meaning that these features are changed over time. Egele et al. [28] build a historical-based model, which does not suffer from this problem. Authors in [29] have built a model using a fuzzy model that attempts to adapt the features over time, but the accuracy is decreased. So, we will focus on this problem and then try to build a robust framework to cope with most of the challenges to detect Twitter spam.

3. Problem Statement

The problem revealed in this paper is detecting and classifying each tweet whether it is spam or not. So, we have the problem of "spam drift", which happened because most of the researchers focus on determining the spam tweets based on the statistical features. Most of them focus on selection of features as shown in Table 1. In the real world, these features are changing in an unpredictable way over time. Therefore, we attempted to build a framework that is robust against these changes.

Table 1. Comparative study of ten consequence days between spam and non-spam using KL-Divergence.

	D-1 vs. D-2		D-2 vs. D-3		D-3 vs. D-4		D-4 vs. D-5		D-5 vs. D-6		D-6 vs. D-7		D-7 vs. D-8		D-8 vs. D-9		D-9 vs. D-10	
F-1	0.37	0.05	0.35	0.04	0.45	0.05	0.25	0.04	0.27	0.04	0.28	0.04	0.30	0.06	0.27	0.04	0.35	0.05
F-2	0.25	0.11	0.23	0.11	0.27	0.11	0.20	0.11	0.22	0.11	0.22	0.11	0.18	0.11	0.39	0.11	0.36	0.11
F-3	0.29	0.08	0.23	0.08	0.33	0.08	0.16	0.08	0.23	0.08	0.21	0.08	0.21	0.09	0.27	0.09	0.24	0.09
F-4	0.17	0.08	0.14	0.08	0.15	0.09	0.15	0.08	0.18	0.08	0.20	0.08	0.14	0.08	0.28	0.09	0.20	0.09
F-5	0.03	0.02	0.03	0.02	0.04	0.02	0.03	0.02	0.02	0.02	0.03	0.02	0.02	0.02	0.06	0.02	0.06	0.02
F-6	0.99	0.36	0.53	0.36	0.64	0.36	0.37	0.36	0.46	0.35	0.41	0.35	0.46	0.36	0.51	0.36	0.53	0.37
F-7	0.11	0.05	0.09	0.04	0.05	0.05	0.05	0.05	0.06	0.04	0.08	0.05	0.07	0.05	0.11	0.05	0.09	0.05
F-8	0.20	0	0	0	0.05	0	0.04	0	0.03	0	0.04	0	0.02	0	0.05	0	0.03	0
F-9	0.10	0	0.04	0	0.02	0	0.03	0	0.02	0	0.02	0	0	0	0.05	0	0.02	0
F-10	0	0	0.04	0	0.04	0	0.02	0	0.11	0	0	0	0.02	0	0.33	0	0.28	0
F-11	0.27	0.02	0.07	0.02	0.07	0.02	0.12	0.02	0.11	0	0.1	0	0.27	0.02	0.29	0.03	0.21	0.03
F-12	0.05	0	0	0	0.03	0	0.04	0.02	0.04	0	0.05	0	0.05	0	0.47	0	0.47	0

At the beginning, we will try to prove this problem as in [29]. So, we have crawled data of tweets from Twitter Stream API for 10 consecutive days. We have to check a lot of tweets to determine which are spam. In this stage, we found that most of the spam tweets contain a URL, which most spammers use to spread their malicious content by sending the victim to mine or farm sites. Therefore, we use Trend Micro's Web Reputation Technology (WRT) to detect the tweet as spam or not based on the URL [22]. This WRT system helps users to identify the malicious sites in real-time with high reliability with an accuracy rate of 100% as reported in [30]. Moreover, we have made hundreds of manual inspections to ensure the reliability of this system.

As described previously, we found that the statistical features are changing from day to day with impressive effect as shown in Table 1. For example, we found that the average number of account followings changes from the 1st day (500–900) to the 9th day (950–1350). This means that the spammers try to collect the followings, but the average number of followings is confused whether this account is spam or not.

Therefore, to justify the problem of changing the statistical features, the distribution of the data should be modeled. There are two types of data: parametric and non-parametric. The parametric approaches are always used when the distribution of data is known as normal distribution, but the statistical features of Twitter are unknown [31,32]. So, we used

the non-parametric approaches. One of the most common non-parametric approaches is the statistical test. The calculation of the statistical test is based on computing the distance between the two distributions to calculate the change between them. Distance is calculated using Kullback-Leibler (KL) divergence [31], which is also known as relative entropy, shown in Equation (1):

$$D_{kl}(P|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (1)$$

This formula is used to measure the two probability distributions as reported in [33]. Let $s = \{x_1, x_2, \ldots, x_n\}$ be a multi-set from a finite set F containing numerical feature values, and $N(x|s)$ is the number of appearances of $x \in s$. Thus, the relative proportion of each x is shown Equation (2)

$$P_s(x) = \frac{N(x|s)}{n} \quad (2)$$

The ratio of the two variables P/Q is undefined, if we assume $Q(i) = 0$. Therefore, the estimation of $P_s(x)$ is changed to Equation (3)

$$P_s(x) = \frac{N(x|s) + 0.5}{n + |F|/2} \quad (3)$$

when variable $|F|$ is defined as the number of elements in the finite set F. The distance between two day's tweets, $D1$ and $D2$, is defined as shown in Equation (4)

$$D(D1|D2) = \sum_{x \in F} P_{D1}(x) \log \frac{P_{D1}(x)}{P_{D2}(x)} \quad (4)$$

We calculate the KL Divergence of spam and legitimate tweets of each feature in two adjacent days as shown in Table 1. The larger the distance, the more dissimilarity between the two distributions. So, according to the results in Table 1, the distance is large in most features in case of spam data. However, in non-spam data, the distance is very small in most of the features. According to this study, by examining the Number_of_tweet (f-6) feature from Table 1, we notice that the KL Divergence metric of spam tweets for Day 1 and Day 2 is 0.99. However, in non-spam tweets, it is 0.36, which means that the distribution of this feature is changed from Day 1 to Day 2 compared to non-spam tweets. As shown in Table 1, most features are changing unpredictably from one day to another, although the training data is fixed and is not affected by any changes. Therefore, the performance of the classifiers will become inaccurate if the decision boundary is not updated.

4. The Proposed Model

The process of classifying tweets as spam or not has three challenges. First, the tweet classification process can not only depend on statistical features because it drifts over time as described. So, our classifier considers the tweet content. Second, our proposed framework must struggle over the spammers because they try to change the tweet content, which helps them to evade from any monitor system [34]. Therefore, new spam tweets must be rephrased from the detected spam. Third, a robust framework must be built that is able to detect spam tweets in less execution time to cope with Twitter big data challenges. These three challenges motivated us to build the proposed framework. This framework consists of three layers as shown in Figure 2.

4.1. Learning from Detected Spam Tweets

This layer is used to filter Twitter as an initial step for fast detection of spam tweets. As described in Figure 2, our proposed framework is interested in spam tweets to regenerate a new semantic meaning of the same tweet by the next layer. Therefore, new information or words can be obtained that the spammer can use to paraphrase the tweet content and spread their spam again. In this step, the SVM classifier is utilized. First, this classifier is

trained with a bi-gram (set of two words for each tweet) and transforms the tweets with TF-IDF. Then, the new unlabeled tweets are entered into this classifier to classify them (Spam, notSpam). This method focuses on the non-spam tweets, which will be the input for the next layer.

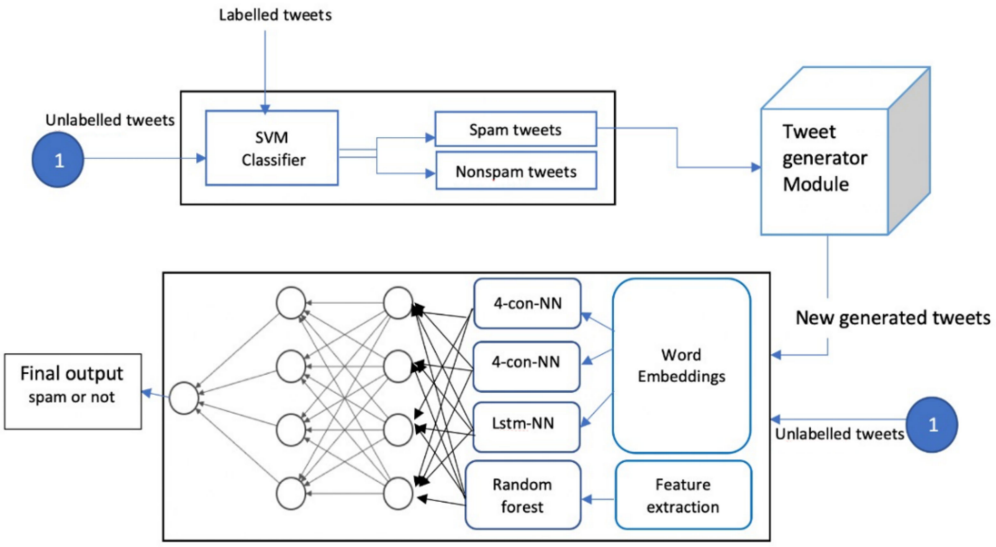

Figure 2. The proposed framework.

4.2. Generate New Tweets

In the real world, researchers try to build robust systems. However, smarter spammers are trying to tackle these solutions. Therefore, a system for tweet paraphrasing should be built using a method that generates text by preserving the same meaning and semantic, not only focusing on the correct grammar. Therefore, we used the encoder-decoder framework [35], which is embedded with an attention model network. The spam tweet paraphrasing model is shown in Figure 3.

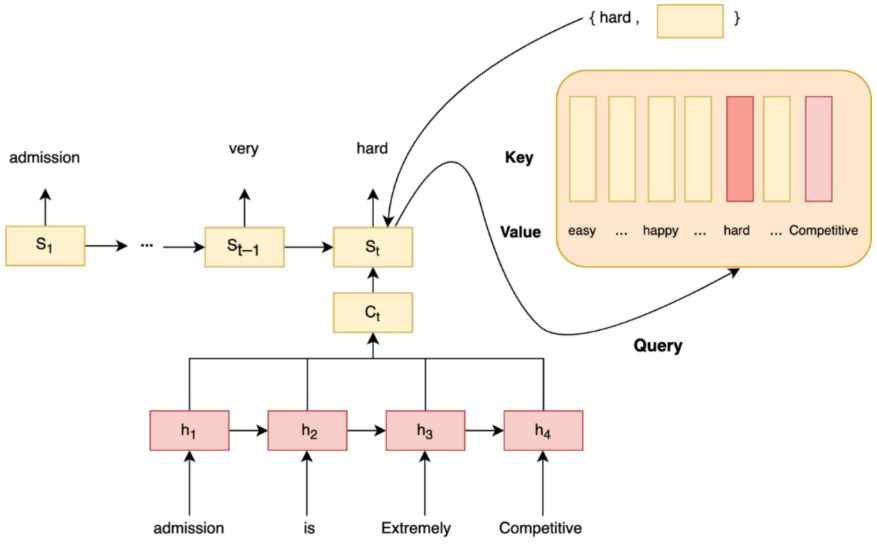

Figure 3. An overview of spam tweet paraphrasing model.

Given the source spam messages as input from the classifier layer, the encoder packs the source into dense representation vectors called context vector c_t , which captures the

context information for this message. Then, the decoder tries to generate the paraphrased messages from the hot encoded vectors according to Equations (5) and (6).

$$c_t = \sum_{i=1}^{N} \alpha_{ti} h_i \tag{5}$$

$$\alpha_{ti} = \frac{e^{g(s_t, h_i)}}{\sum_{j=1}^{N} e^{g(s_t, h_j)}} \tag{6}$$

where $g(s_t, h_i)$ is an attractive score between the encoder state h_i and the decoder state s_t. Then, the dense representations are fed into an attention layer. For predicting words, the decoder utilizes the combination of source and target context vector as query q_t shown in Equation (7) to get the word embeddings

$$q_t = \tanh(W_c[s_t; c_t]) \tag{7}$$

The candidate words W_i and its corresponding embedding vector e_i are stored as key-value pairs $\{e_i, W_i\}$. Therefore, our model uses q_t to query these key-value pairs by evaluating all the applicant words between the query q_t and the word vector W_i as shown in Equation (8)

$$f(q_t, e_i) = \begin{cases} q_t^T e_i \\ q_t^T W_a e_i \\ v^T \tanh(W_q q_t + W_e e_i) \end{cases} \tag{8}$$

where W_q and W_e are two trainable parameter matrices, and v^T is a trainable parameter vector. Then return the word which has the highest matching. The chosen word is emitted as the generated token, and its embedding is then utilized as the contribution of the long short-term memory (LSTM) at the next step. The word embedding is affected by three sources: the input of the encoder, the input of the decoder, and the query of the output layer. In the training stage, we used the Adam optimizer method with these hyper-parameters $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\alpha = 0.001$, and $\in = 1 \times 10^{-8}$

4.3. Ensemble Method

In this layer, we proposed a novel technique to classify the tweets as spam or non-spam as shown in Figure 4. We have combined three deep neural network classifier techniques together for content based on one classifier for user-based features, which contains two different architectures. First, we will explain the methodology for each component and then explain the whole technique as an ensemble classifier.

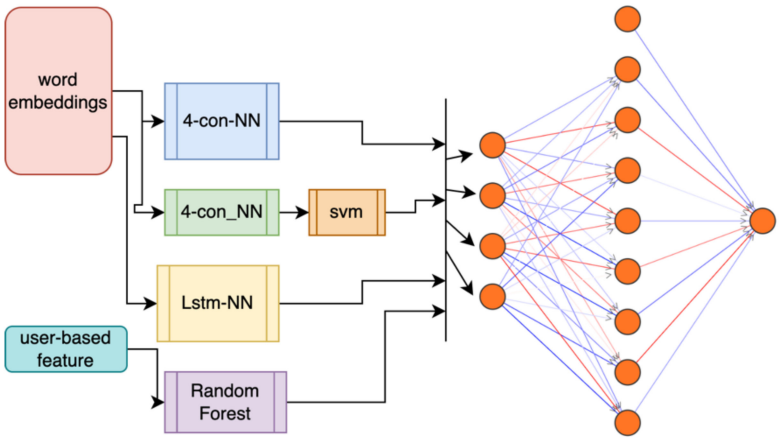

Figure 4. Ensemble neural network architecture.

4.3.1. Convolution Neural Network

In this section, the convolution neural network (CNN) will be discussed. Recently, this network was designed to be used in computer vision problems. However, it has been shown that it can be used in natural language processing (NLP) tasks as [36] proposed neural architecture used in many NLP tasks, such as part of speech tagging, chunk, and named entity recognition. Our model is inspired by [36] in that the layers of this architecture are divided into five parts, input layer, embedding layer, convolution layer, pooling layer, and output layer, as shown in Figure 5. The input layer receives tweet messages as words or embedding words using word2vec or glove [37]. Each tweet is split into words with max_length value 50 because the length of max tweet message is 280 characters, which is difficult to exceed this number of words. If the length is small, it should be padded with value 0. Thereafter, these words are split into features by performing kernel multiplication and then are fed into the next layer, the convolution layer. ReLU, sigmoid, and tanh activation functions are used to obtain the convolution feature map. Then, max pooling is used to select the maximum activation value. Max pooling is used with NLP tasks where min and mean pool is used with computer vision tasks. The fully connected hidden dense layer with sigmoid activation function is applied to classify the tweets. Twelve regularization is used to avoid overfitting. To build this architecture, we used loss function: binary cross entropy and optimizer parameters.

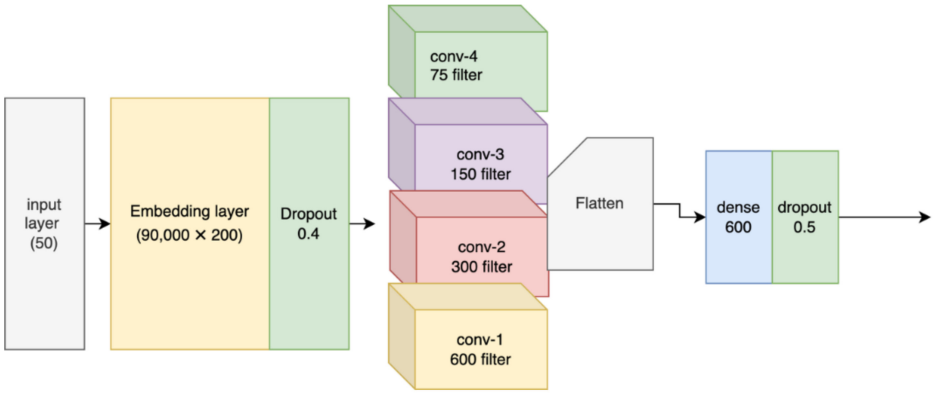

Figure 5. Neural network architecture with four conv. layers.

4.3.2. Recurrent Neural Networks

A recurrent neural network (RNN) is a network of directed connection between each node. The main feature of this network is the hidden state (memory) that can capture the sequential dependence in data. So, we utilized LSTM networks in our work rather than gated recurrent unit (GRU) [38], which has a problem with remembering long sequences. As shown in Figure 6, we used the same architecture as CNN, but we replaced the convolution layer with the LSTM layer which contains three main gates as follows: Forget gate is responsible for controlling what information should throw away from memory, Input gate is responsible for controlling what new information should be added to hidden state from the current input, Output gate decides what information to output from the memory. Then the output of this layer is entered to fully connected dense layer to produce the output.

4.3.3. Feature-Based Model

Statistical features in spam classifiers detection give good results [8]. Apart from using word embedding as described in the previous two sections, we also consider user-based features in our classifier.

A dataset with 6 million tweets is used to extract these features especially for user-based features [29]. We have presented the extracted features that can differentiate between spam or legitimate users as shown in Table 2. To represent the behavior of spam and legitimate accounts, a comparative study has been built between each extracted feature to

represent the difference between them using the empirical cumulative distribution function (ECDF) as shown in Figure 7.

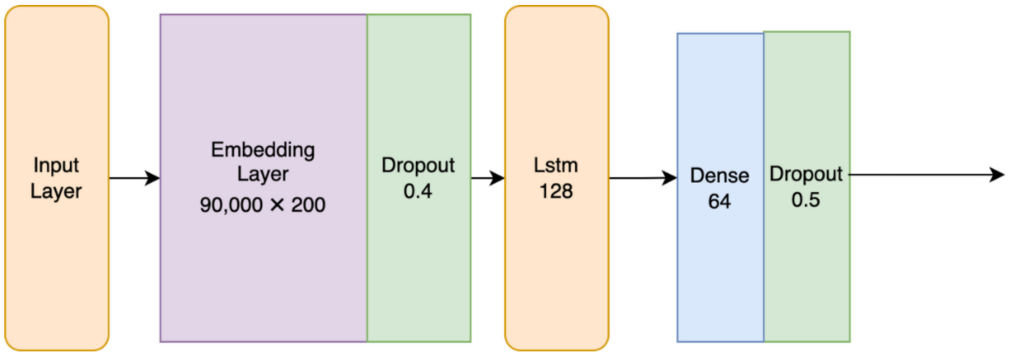

Figure 6. Recurrent neural network architecture.

Table 2. Extracted Features with the Corresponding Description.

Feature No.	Title	Description
F1	Age of account	The count of days of an account from the creation date until the last posted tweet
F2	Number of followers	The count of followers of this Twitter account
F3	Number of followings	The count of friends of this Twitter account
F4	Number of user favorites	The count of favorites this Twitter account added
F5	Number of lists	The size of lists this Twitter account added
F6	Number of tweets	The count of tweets this Twitter account post
F7	Number of retweets	The size of retweets for each tweet
F8	Number of hashtags	The count of hashtags added in this tweet
F9	Number of URLs	The count of user mentions added in this tweet
F10	Number of chars	The count of URLs added in this tweet
F11	Number of digits	The size of characters in this tweet
F12	Number of user mentions	The count of user mentions added in this tweet

The experimental study found that more than 53% of spam users have less than 500-day account age. However, 38% of non-spammers have less than 500 days. This means that they always try to create new accounts to spread their attacks, but they get blocked by spam detection techniques. Also, regarding the number of user mentions, most of the spammers must put more than one user mention to spread their data. Regarding number of capital words, most of the spammers use capital words to attract the users, and more than 70% of spammers use capital words in their tweets compared to only 30% of non-spammer users. In addition, we have also identified a new attribute called reputation of users, which is calculated as shown in Equation (9):

$$\text{Reputation} = \frac{\text{number of followers}}{\text{number of followers} + \text{number of followings}} \quad (9)$$

However, we found that the ratio of spammers is always small. They always have number of followings more than number of the followers because they try to make fake followers or following to show that this is a real account.

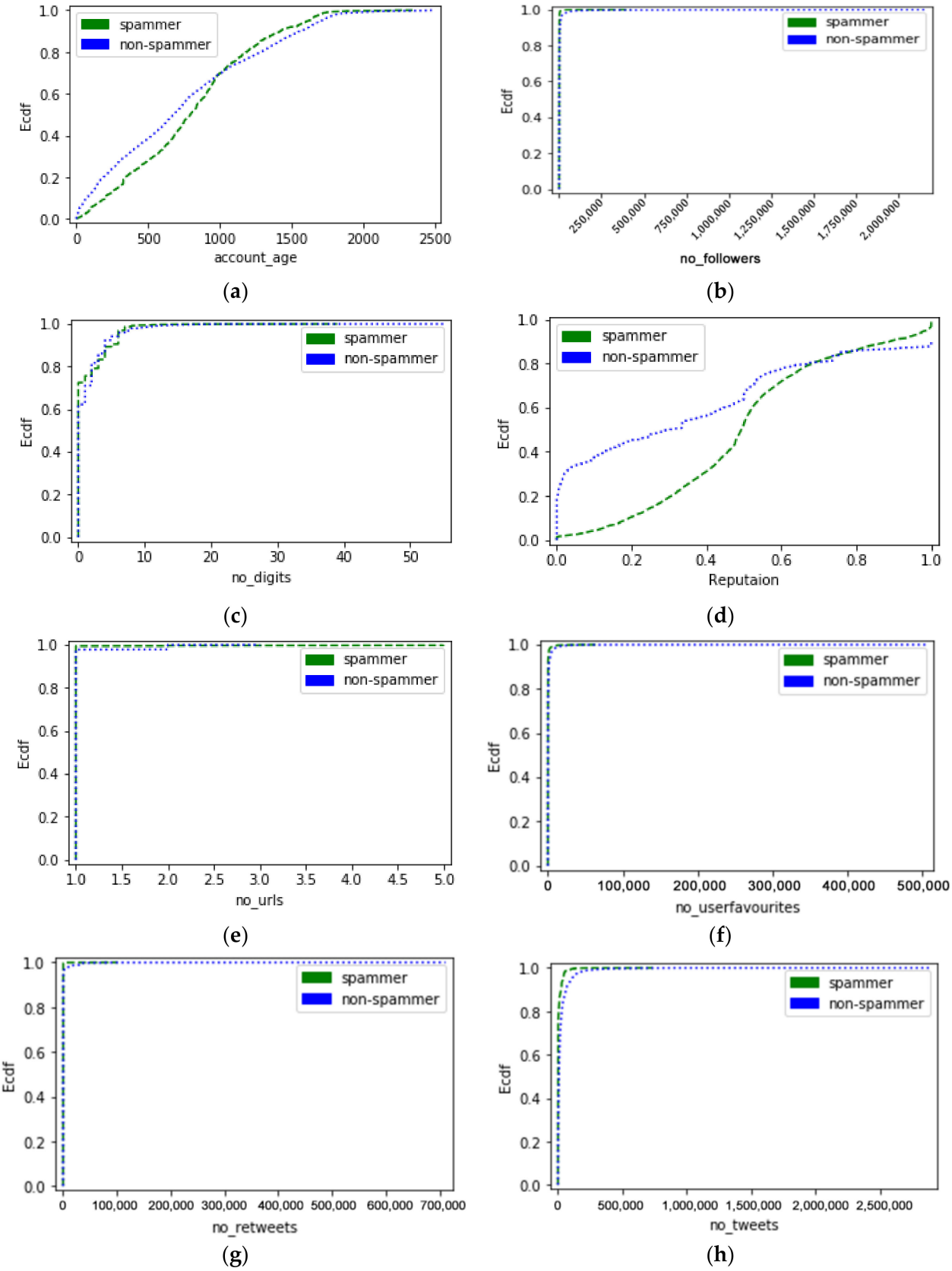

Figure 7. *Cont*.

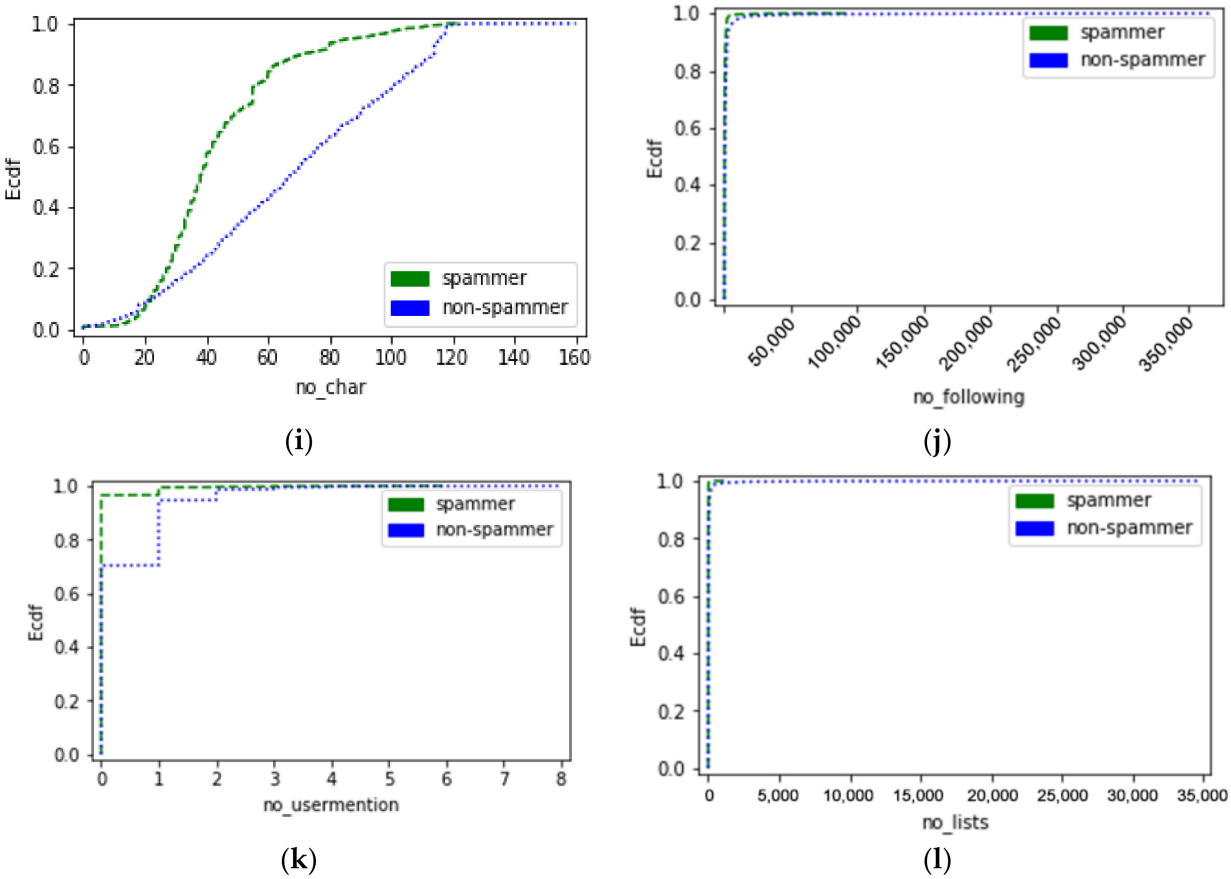

Figure 7. (ECDF) User-based features comparison: (**a**) account age; (**b**) number of followers; (**c**) number of digits; (**d**) reputation; (**e**) number of URLs; (**f**) number of user favorites; (**g**) number of retweets; (**h**) number of tweets; (**i**) number of characters; (**j**) number of followings; (**k**) number of user mentions; (**l**) number of lists.

4.3.4. Proposed Ensemble Approach

As shown in Figure 4, this architecture contains three different neural networks gathered with one classifier for a user-based feature and is described as follows:

- First, CNN is used with four convolution layers, which is trained with Twitter glove [37].
- Second, CNN is also used with four convolution layers to extract features and then classify them using the SVM algorithm. This CNN is trained with Twitter Glove in all dimensions.
- Third, the LSTM network is used and trained with the Hspam dataset, which contains 14 million tweets [18] and with Twitter Glove.
- Finally, random forest is used to classify the user-based features as it gives the best results according to [8,39]. It is trained using the icc dataset [40].

Furthermore, a neural network meta classifier is utilized and trained from the newly created data which consists of three-layers. It contains four input nodes and eight hidden nodes with a bias that is supported with the ReLU activation function. The output has only one node supported with the sigmoid activation function to generate value from 0 to 1.

5. Experiments and Results

In this section, we will present our experiments for each approach with different datasets for detecting the spam tweets in the Twitter platform. Firstly, we will give a brief description of our datasets and the evaluation metrics used in this study, then we will discuss our results of each approach.

5.1. Dataset

A ground truth dataset, which is called Hspam, is applied [18]. It contains 14 million tweets collected over two months and classified using many methods, such as manual annotation, KNN-based annotation, user-based annotation, domain-based annotation, and reliable ham tweet detection. For the privacy of the Twitter platform, we must grab the tweets using tweet_id, but there are some tweets that are deleted or missed. So, we focus only on the returned tweets. To evaluate our approaches over many datasets, we split our dataset into 4 samples as shown in Table 3. We made two balanced samples with random selection and another with continuous selection. Then, we selected another two samples and divided the ratio of spam to not spam to 20 times as it describes that, in real life, 5% only of tweets are spam [6]. So, we made two samples to simulate the real-life data. For testing our approaches, we selected a random sample of 0.5 million tweets to make a fair comparison between all dataset samples and all approaches.

Table 3. Dataset Samples.

Dataset No.	Type	Spam:Not-Spam
1	random	200 k:200 k
2	continuous	200 k:200 k
3	random	50 k:1000 k
4	continuous	50 k:1000 k

5.2. Evaluation Metrics

To evaluate our approach, we used the metrics of recall, precision, and F1-score which are shown in Equations (10)–(12), respectively. We supposed that spam tweets are positive while non-spam tweets are negative. Then, we constructed the confusion matrix accordingly as shown in Table 4, where TP (true-positive) refers to all spam tweets that are predicted correctly as spam tweets, FN (false-negative) denoted as all spam tweets which are predicted wrongly as non-spam tweets, TN (true-negative) denoted as all non-spam tweets which are predicted correctly as non-spam tweets, and FP (false-positive), which refers to all non-spam tweets predicted wrongly as spam tweets.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{11}$$

$$F1 - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

Table 4. Confusion Matrix.

		Predicted	
		spam	Not-spam
Actual	spam	TP	FP
	Not-spam	FN	TN

5.3. Experiments Settings

We have run our experiments in Linux ubuntu 18 LTS, with Inter(R) core (TM) I7 CPU of 16 GB. For each run over each dataset with every model, we divide the dataset into 80% as a training set and 20% for testing. All basic parameters we use in each model are

put in each figure in the last section, embedding layer, dropout, number of filters, and dense network.

6. Results and Discussion

In this subsection, we will discuss the results of each model in our proposed framework and compare it with the latest frameworks.

6.1. Primary Twitter Filter

In this section, maxentropy, random forest, and SVM are implemented. As shown in Table 5 and Figure 8, SVM achieved the best results in terms of recall, precision, and F1-score for most datasets. So, it is selected to be applied in our framework with parameters c = 0.1, kernel = linear, and penalty = 12.

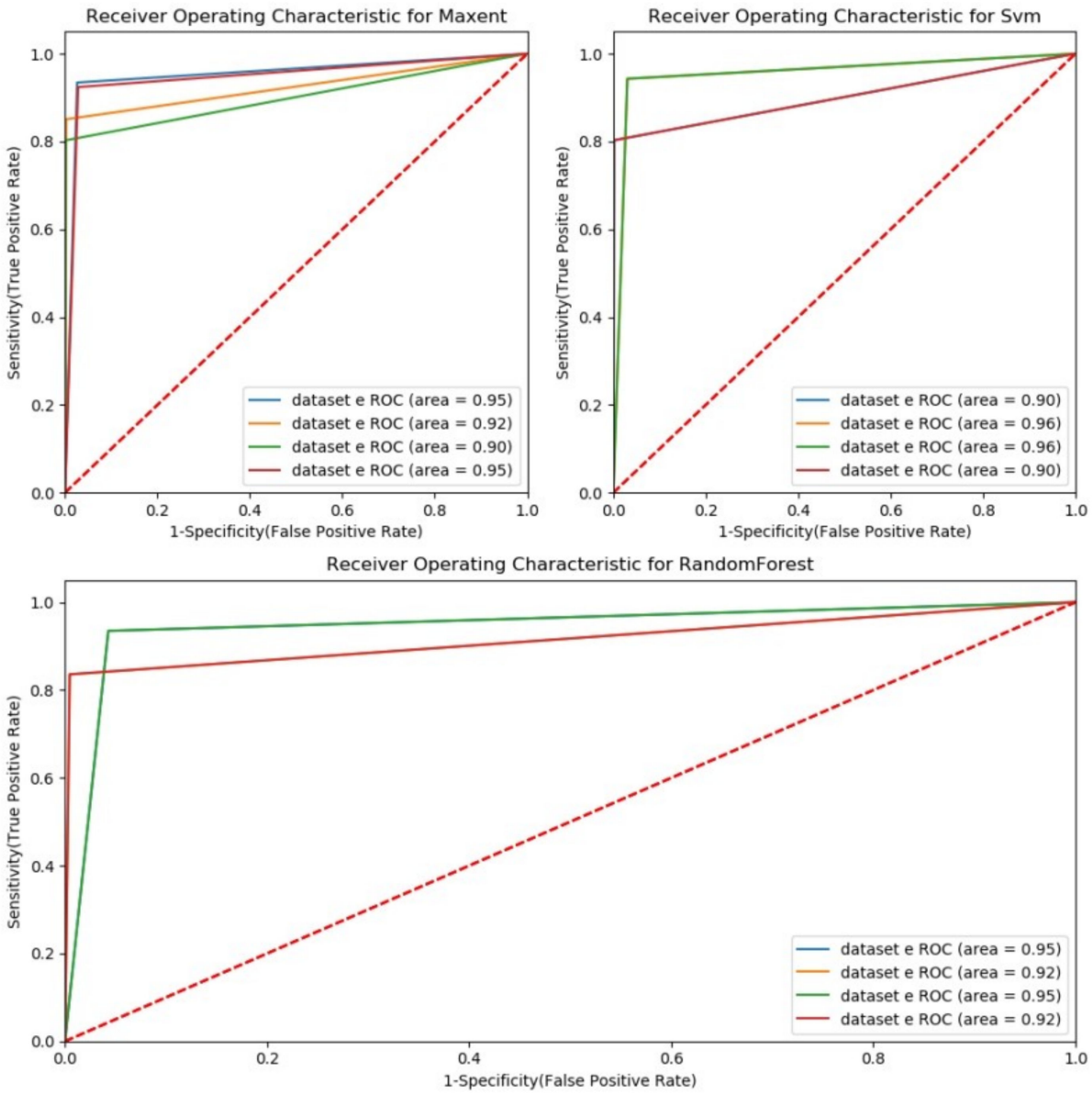

Figure 8. Roc curve for comparative study for SVM, MaxEntropy and Random Forest algorithms foreach dataset as the first module for filtering the tweets.

Table 5. Evaluation Results for Dataset 1.

Method		Precision	Recall	F-Measure
First module	MaxEntropy	0.96	0.95	0.95
	RandomForest	0.96	0.95	0.95
	SVM	0.97	0.96	0.96
LSTM		0.95	0.96	0.95
CNN		0.92	0.95	0.93
CNN + SVM		0.95	0.95	0.95
Random Forest (user-based feature)		0.96	0.90	0.93
SVM (user-based feature)		0.94	0.84	0.89
Chen et al. [41]		0.85	0.64	0.73
Wang et al. [40]		0.94	0.80	0.86
Madisetty et al. [42]		0.94	0.95	0.94
Proposed method		0.96	0.96	0.96

6.2. User-Based Features

As discussed earlier, the statistical features are changed over time, but this cannot prevent their abilities to detect spammers' actions with high accuracy and precision. Therefore, we attempt to find new user-based features. SVM and random forest are compared to get the best algorithm to be part of our detection framework. As shown in Tables 5 and 6, random forest achieves the best results in terms of precision and recall where trained with 6 million-tweet dataset [40] to get the user-based statistical features.

Table 6. Evaluation Results for Dataset 4.

Method		Precision	Recall	F-Measure
First module	MaxEntropy	0.96	0.95	0.95
	RandomForest	0.96	0.95	0.95
	SVM	0.96	0.96	0.96
LSTM		0.98	0.93	0.95
CNN		0.95	0.89	0.93
CNN + SVM		0.97	0.89	0.93
Random Forest (user-based feature)		0.60	0.70	0.65
Chen et al. [41]		0.58	0.67	0.62
Wang et al. [40]		0.79	0.76	0.77
Madisetty et al. [42]		0.92	0.94	0.93
Proposed method		0.97	0.95	0.96

6.3. Ensemble Method

This is the main module that consists of three main algorithms as discussed previously. They are trained with the Twitter Glove word embedding [37] dataset for all dimensions 25, 50, 100, 200. The results of each dimension are compared to our four datasets for each model as shown in Figures 9–11. We found that the results for the 200 dimensions are better in the three models, CNN, LSTM, and CNN with SVM.

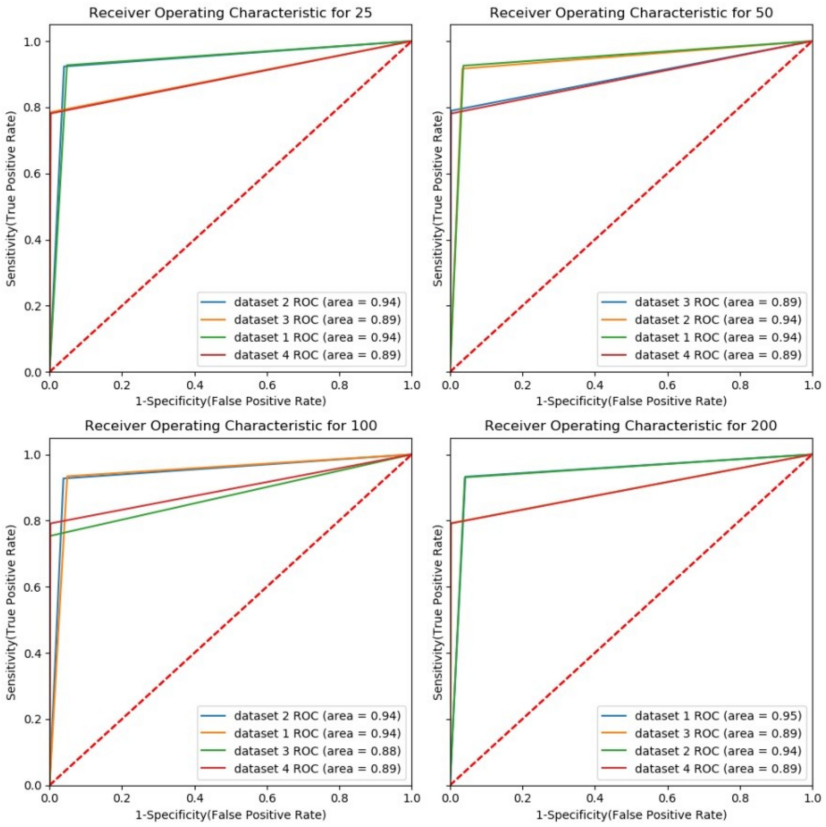

Figure 9. Roc curve for CNN model results for each dataset as a first component in our ensemble method.

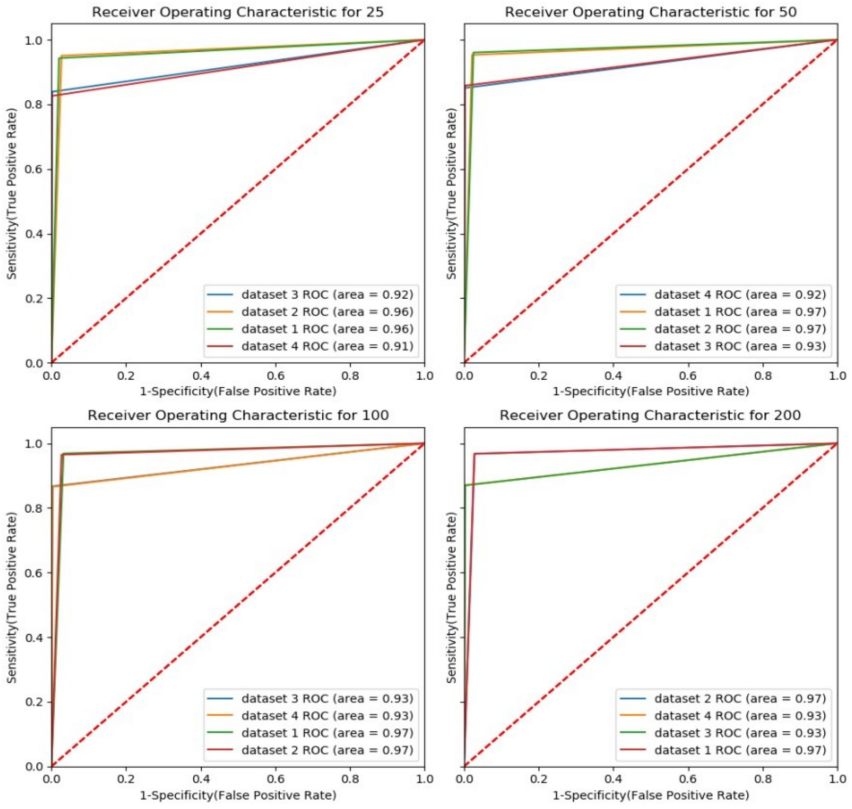

Figure 10. ROC curve for LSTM model results for each dataset as a third component in our ensembl method.

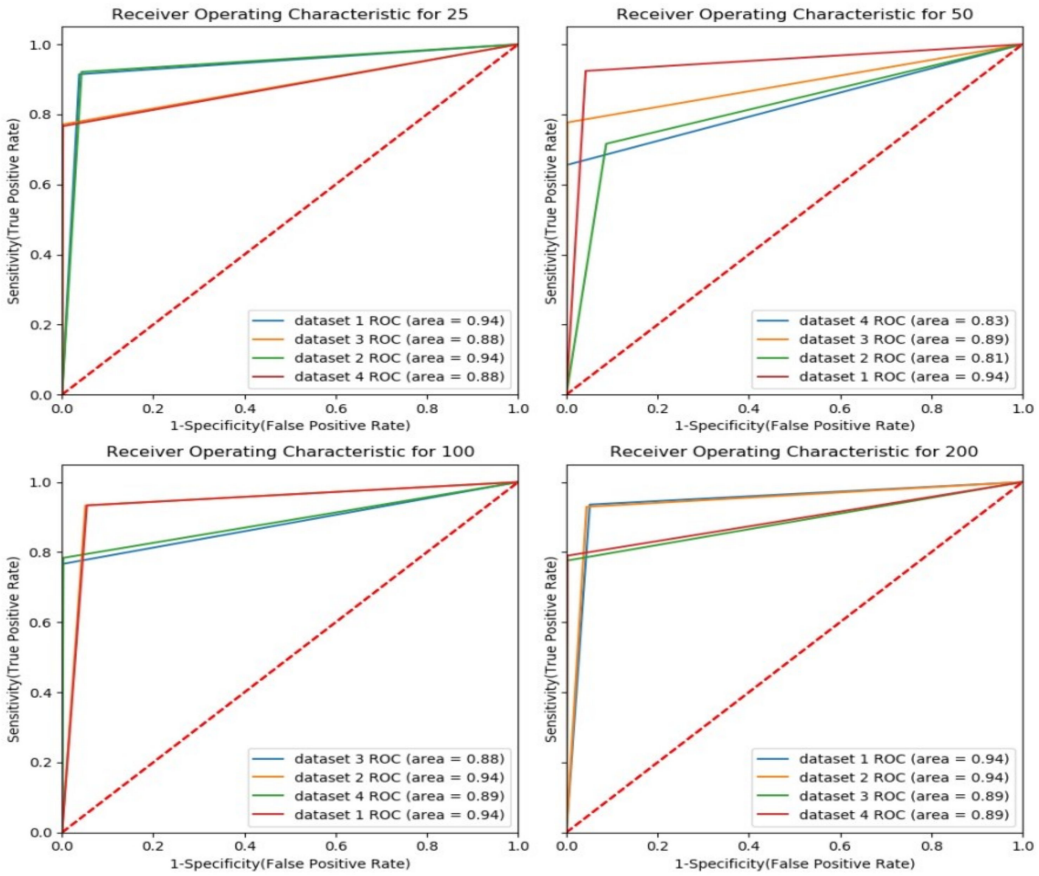

Figure 11. ROC curve for CNN features with SVM model results for each dataset as a second component in our ensemble method.

The CNN model is very good at finding the patterns. Each convolution will fire when a learned pattern is detected, but it suffers from long patterns or long tweets, which make the results of precision and F1-measure less. So, we embedded that LSTM model that is built using RNN, which is the strongest one with long sequences compared to CNN. Some studies [43] conducted an alternative for the last softmax function by SVM model. It aims to decide the optimal hyperplane for isolating the two classes in the dataset, and a multinomial case is apparently disregarded. With the utilization of SVM in a multinomial classification, the case turns into a one-versus-all, in which the positive class has the highest score, while the rest has the negative class.

6.4. Meta-Classifier

To achieve the results of our proposed framework, we build a sequential neural network that assembles the results of the utilized methods: LSTM, CNN, CNN feats with SVM, user-based features as presented in Figure 12. As shown in Tables 5 and 6, the proposed model achieved the best results in terms of accuracy, precision, and recall compared to the latest research in this field. Although [41] has the lowest execution time that it take 0.002 for each tweet, this execution time is very small compared to our proposed method as it takes longer, approximately 2 ms for each tweet. That is because of the number of features used to detect the spam and the combination of models that the tweet must pass to get the final result. However, this time can be optimized using clusters of nodes to decrease the time.

We also found that the results of the meta classifier are not boosted very much as they are too close to the ensemble model, but it is able to preserve the performance by a significant margin for this dataset. So, we can offer robust framework, that can be self-

trained with the new words and hashtags, which the spammer can use as Twitter always has new subjects and interests of their users.

6.5. Performance of Learned Model

Twitter is considered a real-time platform. Therefore, it is extremely important to block spam tweets before it spread for preserving the safety of its users and preventing any potential damage. So, the proposed framework is designed to observe the execution time of the detection process. The processing time is calculated for the whole framework for each tweet. We found that each tweet takes 1:2 ms to detect whether it is spam or not. This value is very acceptable in real-time applications, although it can be decreased by using clusters of these models that help with the parallelization of the execution of the process of detecting spamming activities. However, most of the spammers are always thinking out of the box.

They try to deceive all detection strategies by changing the keywords and content and trying new features that can pass from detection methods and attract the users. On the other hand, there are legitimate users who are posting in new trending topics and new events happened immediately. So, we need to retrain the detection framework periodically to preserve the same accuracy and performance which we added in designing our framework, while all systems that depend only on the statistical features will be useless at later time. Our framework combined the statistical features with the deep learning features. So, it is very difficult for a spammer to fool our detection system. Furthermore, we have conducted four experiments with different datasets to test our framework. We concluded that our framework gives good results in both balanced and imbalanced datasets where the imbalanced dataset 4 has 1 million tweets and the balanced dataset 1 has 0.4 million tweets. They gave the same results in precision and F1-measure, which show the robustness of our detection framework as shown in Tables 5 and 6.

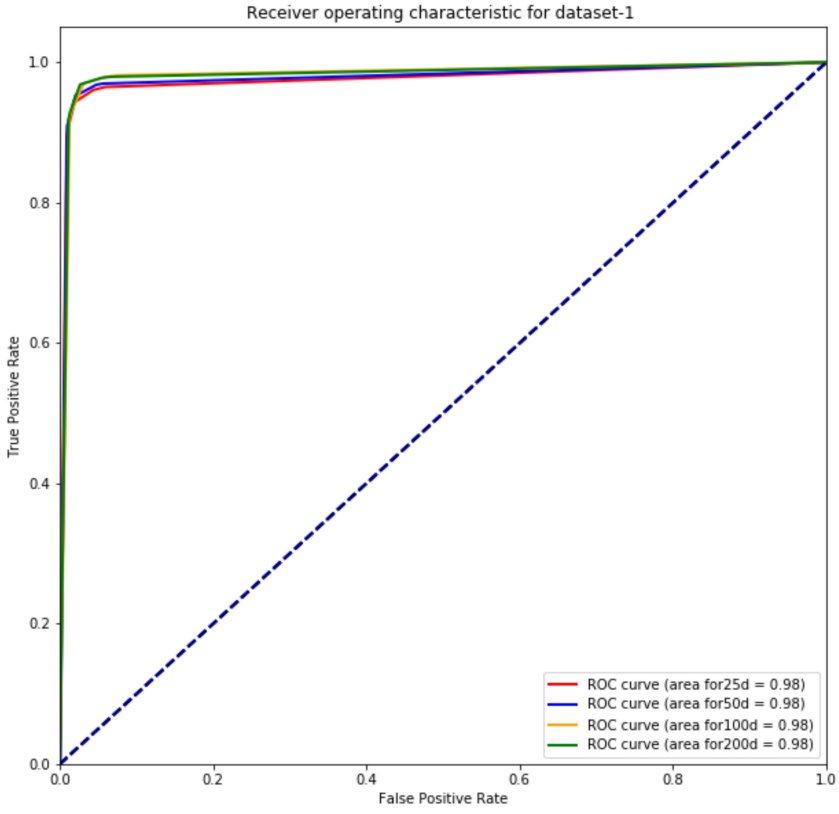

Figure 12. *Cont.*

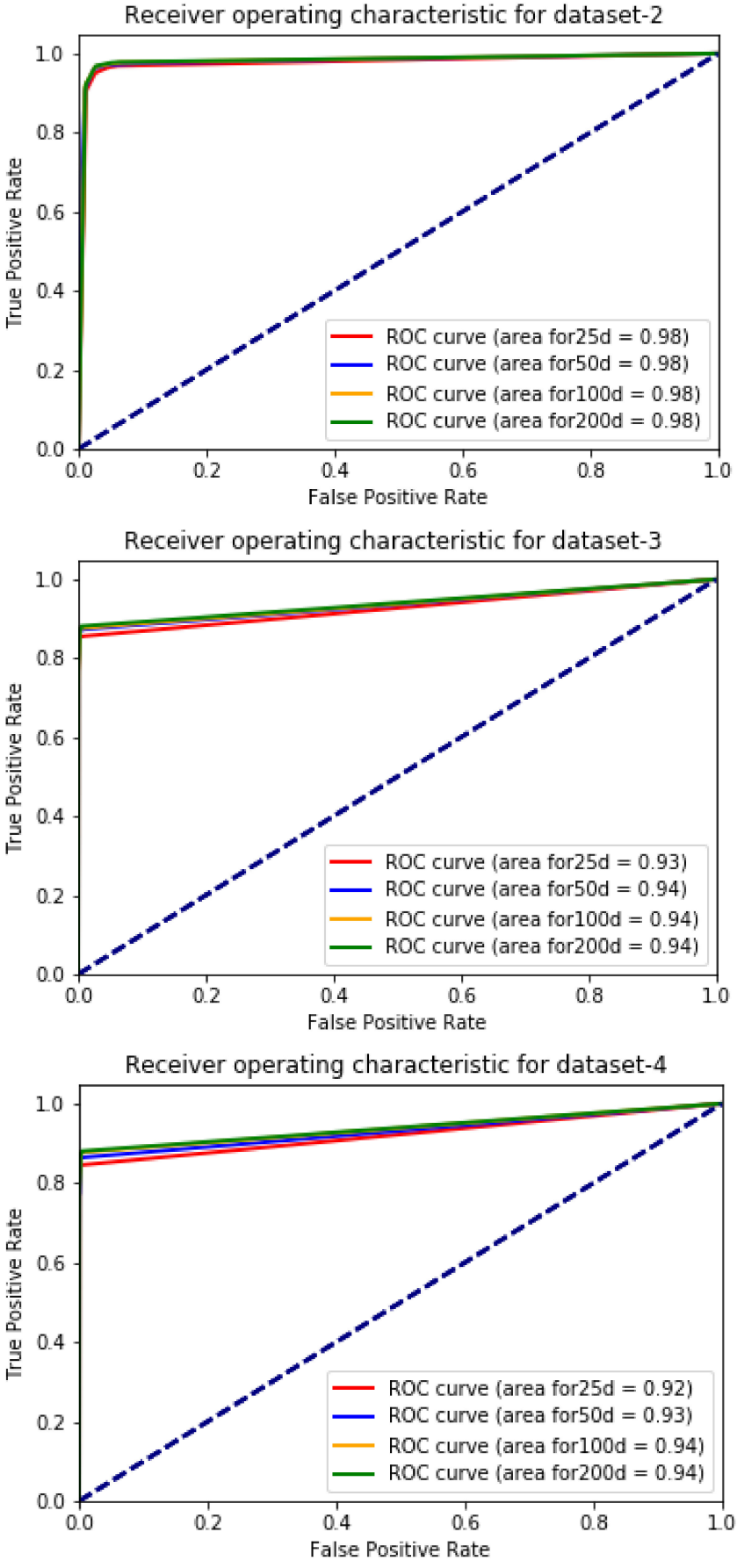

Figure 12. ROC curve for the results of our proposed framework for different datasets 1, 2, 3, and 4.

7. Conclusions

In this paper, we have proposed an ensemble learning framework based on deep learning technique that tries to detect spam tweets based on two methods: firstly, working at the tweet level by building three robust models; secondly, work with a user-based feature to gather information between the user information and the words in each tweet. We also tried to get ahead of step by generating new spam tweets to train our models to predict any spam paraphrasing those spammers can try to deceive our users. The proposed model has been trained using four datasets for more than 7 million tweets to build a robust framework. The experiments show that our proposed model gives excellent results compared to other methods in an acceptable time.

In future work, we will try to conduct more experiments in other online social networks rather than Twitter. Also, we will consider other data formats, such as images and videos that can affect OSN platforms. In addition, we need to try our model in new real data to study if our framework can be affected by the changing of data.

Author Contributions: Conceptualization, A.A. and M.M.; methodology, A.A.; software, M.M.; validation, A.A. and M.M.; formal analysis, M.M.; investigation, A.A.; resources, A.A.; data curation, M.M.; writing—original draft preparation, M.M.; writing—review and editing, A.A.; visualization, M.M.; supervision, A.A.; project administration, A.A.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported through the Annual Funding track by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Project No. AN000417].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The dataset is available on http://nsclab.org/nsclab/resources/?fbclid=IwAR2SkJQ9hN-0LCTb54UYdBCm7CS10zZqgywrh4lOtJo7M4JxjCr2D184QYk, (accessed on 20 April 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chu, Z.; Widjaja, I.; Wang, H. Detecting social spam campaigns on twitter. In Proceedings of the International Conference on Applied Cryptography and Network Security, Singapore, 26–29 June 2012; pp. 455–472.
2. Ghosh, S.; Viswanath, B.; Kooti, F.; Sharma, N.K.; Korlam, G.; Benevenuto, F.; Ganguly, N.; Gummadi, K.P. Understanding and combating link farming in the twitter social network. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012; pp. 61–70.
3. Adewole, K.S.; Anuar, N.B.; Kamsin, A.; Varathan, K.D.; Razak, S.A. Malicious accounts: Dark of the social networks. *J. Netw. Comput. Appl.* **2017**, *79*, 41–67. [CrossRef]
4. Zhu, Y.; Wang, X.; Zhong, E.; Liu, N.N.; Li, H.; Yang, Q. Discovering spammers in social networks. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012.
5. Lee, S.; Kim, J. Warningbird: A near real-time detection system for suspicious urls in twitter stream. *IEEE Trans. Dependable Secur. Comput.* **2013**, *10*, 183–195. [CrossRef]
6. Grier, C.; Thomas, K.; Paxson, V.; Zhang, M. @ spam: The underground on 140 characters or less. In Proceedings of the 17th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 4–8 October 2010; pp. 27–37.
7. Thomas, K.; Grier, C.; Ma, J.; Paxson, V.; Song, D. Design and evaluation of a real-time url spam filtering service. In Proceedings of the 2011 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 22–25 May 2011; pp. 447–462.
8. Wu, T.; Wen, S.; Xiang, Y.; Zhou, W. Twitter spam detection: Survey of new approaches and comparative study. *Comput. Secur.* **2018**, *76*, 265–284. [CrossRef]
9. Ma, J.; Saul, L.K.; Savage, S.; Voelker, G.M. Learning to detect malicious urls. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–24. [CrossRef]
10. Yardi, S.; Romero, D.; Schoenebeck, G. Detecting spam in a twitter network. *First Monday* **2010**, *15*. [CrossRef]
11. Lee, K.; Caverlee, J.; Webb, S. Uncovering social spammers: Social honeypots+ machine learning. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 435–442.
12. Benevenuto, F.; Magno, G.; Rodrigues, T.; Almeida, V. Detecting spammers on twitter. In Proceedings of the Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS), Redmond, WA, USA, 13–14 July 2010; Volume 6, p. 12.

13. Stringhini, G.; Kruegel, C.; Vigna, G. Detecting spammers on social networks. In Proceedings of the 26th Annual Computer Security Applications Conference, Austin, TX, USA, 6–10 December 2010; pp. 1–9.
14. Wang, A.H. Don't follow me: Spam detection in twitter. In Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT), Athens, Greece, 26–28 July 2010; pp. 1–10.
15. Song, J.; Lee, S.; Kim, J. Spam filtering in twitter using sender-receiver relationship. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Menlo Park, CA, USA, 20–21 September 2011; pp. 301–317.
16. Yang, C.; Harkreader, R.; Gu, G. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1280–1293. [CrossRef]
17. Mostafa, M.; Abdelwahab, A.; Sayed, H.M. Detecting spam campaign in twitter with semantic similarity. *J. Phys. Conf. Ser.* **2020**, *1447*, 12044. [CrossRef]
18. Sedhai, S.; Sun, A. Hspam14: A collection of 14 million tweets for hashtag-oriented spam research. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 223–232.
19. Sedhai, S.; Sun, A. Semi-supervised spam detection in Twitter stream. *IEEE Trans. Comput. Soc. Syst.* **2017**, *5*, 169–175. [CrossRef]
20. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
21. Hosseinalipour, A.; Ghanbarzadeh, R. A novel approach for spam detection using horse herd optimization algorithm. In *Neural Computing & Applications*; Springer: New York, NY, USA, 2022. [CrossRef]
22. Abayomi-Alli, O.; Misra, S.; Abayomi-Alli, A. A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset. In *Concurrency and Computation Practice and Experience*; Wiley: Hoboken, NJ, USA, 2022. [CrossRef]
23. Sitaula, C.; Basnet, A.; Mainali, A.; Shahi, T.B. Deep Learning-Based Methods for Sentiment Analysis on Nepali COVID-19-Related Tweets. *Comput. Intell. Neurosci.* **2021**, *2021*, 2158184. [CrossRef]
24. Shahi, T.B.; Sitaula, C.; Paudel, N. A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification. *Comput. Intell. Neurosci.* **2022**, *2022*, 5681574. [CrossRef]
25. Aizawa, A. An information-theoretic perspective of TF–IDF measures. *Inf. Process. Manag.* **2003**, *39*, 45–65. [CrossRef]
26. Church, K.W. Word2Vec. *Nat. Lang. Eng.* **2017**, *23*, 155–162. [CrossRef]
27. Fei, S.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1441–1450.
28. Egele, M.; Stringhini, G.; Kruegel, C.; Vigna, G. Toward detecting compromised accounts on social networks. *IEEE Trans. Dependable Secure Comput.* **2017**, *14*, 447–460. [CrossRef]
29. Chen, C.; Wang, Y.; Zhang, J.; Xiang, Y.; Zhou, W.; Min, G. Statistical features-based real-time detection of drifted twitter spam. *IEEE Trans. Inf. Forensics Secur.* **2016**, *12*, 914–925. [CrossRef]
30. Whole Product Dynamic Real-World Protection Test. 2016. Available online: https://www.av-comparatives.org/testmethod/real-world-protection-tests/ (accessed on 12 August 2020).
31. Dasu, T.; Krishnan, S.; Venkatasubramanian, S.; Yi, K. An information-theoretic approach to detecting changes in multi-dimensional data streams. In Proceedings of the Symposium on the Interface of Statistics, Computing Science, and Applications, Pasadena, CA, USA, 24–27 May 2006.
32. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv.* **2014**, *46*, 1–37. [CrossRef]
33. Csiszar, I.; Körner, J. *Information Theory: Coding Theorems for Discrete Memoryless Systems*; Cambridge University Press: Cambridge, UK, 2011.
34. Chen, C.; Zhang, J.; Xiang, Y.; Zhou, W.; Oliver, J. Spammers are becoming "Smarter" on Twitter. *IT Prof.* **2016**, *18*, 66–70. [CrossRef]
35. Ma, S.; Sun, X.; Li, W.; Li, S.; Li, W.; Ren, X. Query and output: Generating words by querying distributed word representations for paraphrase generation. *arXiv* **2018**, arXiv:1803.01465.
36. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
37. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global vectors for word representation. In Proceedings of the EMNLP 2014—2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
38. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
39. Verma, M.; Sofat, S. Techniques to detect spammers in twitter—A survey. *Int. J. Comput. Appl.* **2014**, *85*. [CrossRef]
40. Zhang, J.; Chen, C.; Chen, X.; Xiang, Y.; Zhou, W. 6 million spam tweets: A large ground truth for timely Twitter spam detection. In Proceedings of the IEEE International Conference on Communications, London, UK, 8–12 June 2015; pp. 7065–7070.
41. Wang, B.; Zubiaga, A.; Liakata, M.; Procter, R. Making the most of tweet-inherent features for social spam detection on Twitter. *arXiv* **2015**, arXiv:1503.07405.
42. Madisetty, S.; Desarkar, M.S. A neural network-based ensemble approach for spam detection in Twitter. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 973–984. [CrossRef]
43. Agarap, A.F. An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification. *arXiv* **2017**, arXiv:1712.03541.