processes

MDPI

Article

A Combined Text-Based and Metadata-Based Deep-Learning Framework for the Detection of Spam Accounts on the Social Media Platform Twitter

Atheer S. Alhassun [1,*] and Murad A. Rassam [1,2,*]

[1] Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[2] Faculty of Engineering and Information Technology, Taiz University, Taiz 6803, Yemen
* Correspondence: 411200029@qu.edu.sa (A.S.A.); m.qasem@qu.edu.sa (M.A.R.)

Abstract: Social networks have become an integral part of our daily lives. With their rapid growth, our communication using these networks has only increased as well. Twitter is one of the most popular networks in the Middle East. Similar to other social media platforms, Twitter is vulnerable to spam accounts spreading malicious content. Arab countries are among the most targeted, possibly due to the lack of effective technologies that support the Arabic language. In addition, as a complex language, Arabic has extensive grammar rules and many dialects that present challenges when extracting text data. Innovative methods to combat spam on Twitter have been the subject of many current studies. This paper addressed the issue of detecting spam accounts in Arabic on Twitter by collecting an Arabic dataset that would be suitable for spam detection. The dataset contained data from premium features by using Twitter premium API. Data labeling was conducted by flagging suspended accounts. A combined framework was proposed based on deep-learning methods with several advantages, including more accurate, faster results while demanding less computational resources. Two types of data were used, text-based data with a convolution neural networks (CNN) model and metadata with a simple neural networks model. The output of the two models combined identified accounts as spam or not spam. The results showed that the proposed framework achieved an accuracy of 94.27% with our combined model using premium feature data, and it outperformed the best models tested thus far in the literature.

Keywords: online social network; Arabic spam account; spam detection; deep learning; deep convolution neural networks

Citation: Alhassun, A.S.; Rassam, M.A. A Combined Text-Based and Metadata-Based Deep-Learning Framework for the Detection of Spam Accounts on the Social Media Platform Twitter. *Processes* **2022**, *10*, 439. https://doi.org/10.3390/pr10030439

Academic Editor: Chien-Chih Wang

Received: 26 January 2022
Accepted: 20 February 2022
Published: 22 February 2022

1. Introduction

Currently, online social networks (OSNs) are integral to our lives. Twitter, Facebook, and Instagram are some of the OSNs that have transformed the way we socialize and connect [1]. Every type of information can be publicized to a wide audience within seconds using a social media platform [2].

As one of the most significant online social networks [3], Twitter attracts users by offering a free "microblogging" service for posting short messages named "tweets" [4,5]. Twitter's rapid growth has been the result of users posting millions of tweets [6]. Originally, each tweet was allowed only 140 characters, but recently, Twitter has expanded messages to 280 characters. Texts, URLs, icons, videos, and images may be included in a post [2,7]. Users connect with other users by "following" them, as each account has a public following and follower count, creating the "social" aspect of the social network [7]. Twitter has a unique function referred to as a "mention" where users can publicly tag another user by putting the other user's Twitter handle after the "@" symbol, and Twitter notifies the other user of the mention. In addition, users can "retweet" posts that are shared publicly by other accounts to their own followers, with or without their own commentary. Finally, a

hashtag can be generated by inserting the hash symbol at the start of an unbroken word string inside a post [7,8]. Searching for information of interest or the latest news via Twitter is simple: users can enter hashtags or keywords into the search function, or even just click on a hashtag they see in their feed or in the "trending" section, to review all the posts containing that keyword or hashtag [2].

Twitter generates 500 million tweets daily via its 152 million active users. The number of active users per month is approximately 353 million; 70% are male, and approximately 30% are female [3,9]. According to a study [9], Saudi Arabia and Egypt are among the 20 leading countries based on the number of Twitter users as of October 2020, as shown in Figure 1.

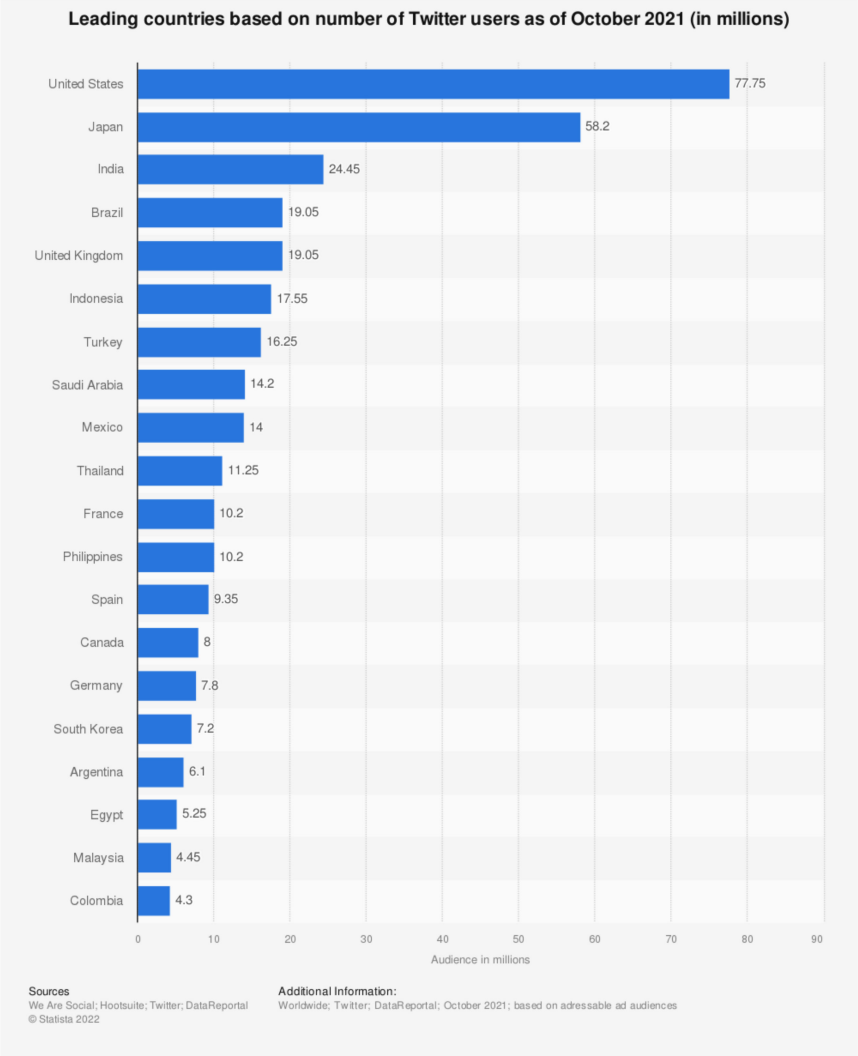

Figure 1. Twitter users' most leading countries (numbers in millions) [9].

This rapid, expansive growth has also made Twitter the optimal environment for spam accounts to spread malicious content. Spam accounts typically use subterfuge to appeal to legitimate users by publishing misleading ads, selling drugs, or embedding subversive, malicious links [1]. Twitter's content-filtering strategies have not been successful due to the adaptations of spam accounts as they attempt to overcome such filters. For example, spam accounts have shared shortened URLs to surpass the 140-character limit established by Twitter. Therefore, part of Twitter's strategy when increasing post lengths to 280 characters was to reduce the use of URL-shortening [5].

Arab countries are one of the most targeted countries by spam accounts, likely due to the lack of technological innovation that provides support for the Arabic language in order

to address these attacks. This may be due to the difficulty of navigating a complex language such as Arabic, which has many different grammatical rules for both the spoken and written forms. One word can have different forms using various suffixes. Using the same three-letter root, several words with vastly different meanings can be produced [10,11]. Moreover, informal Arabic contains multiple dialects that can vary from region to region, all of which are used on social media platforms [2,12]. Therefore, this has made conducting preprocessing operations for spam detection in Arabic a complex and difficult process, particularly due to the frequent use of colloquial words.

Existing approaches depend on the use of machine learning and the extraction of certain types of features from user accounts, posts, or social graphs. These approaches have proven successful; however, they require control from the user, high resources, and more processing time. In recent years, deep-learning methods have surpassed previous approaches in speed and efficiency, and they have not required user intervention [13]. Extracting features from text rather than regular data has been the dominant trend in recent deep-learning-based studies. This is due to the ability of deep-learning algorithms to extract and track hidden patterns within these texts that a traditional approach may not be able to detect or predict [13].

Most researchers in the field of detecting Arabic spam tweets have concentrated on labeling their datasets using two methods. First, they have approached the task manually, where the researcher sets specific rules to determine whether an account was spam or not. Therefore, the dataset has been limited by a single researcher's perspective and experience, which may affect the performance and results. In addition, this method is difficult and time-consuming. Second, researchers have used customized keywords, where a set of words specified by the researcher was used, and each tweet containing one of these words was then classified as spam. However, the use of keywords can vary in different countries and may also drift over time and with changing interests. Therefore, adopting a method where accounts are suspended by Twitter's internal regulations would provide more consistency and accuracy when identifying spam accounts.

The use of traditional machine-learning-based methods that depend on extracting features manually has not been viable. Spam accounts have discovered new methods to bypass the controls established. The use of deep-learning techniques, instead of machine learning, has several advantages including more accurate results and speed, as extracting features does not require manual work or high computational resources.

This paper collected a large dataset of tweets in Arabic in order to develop a novel framework for detecting Arabic spam accounts on Twitter using deep convolutional neural networks. The proposed framework offered two models: the first was based on text data only, and the second combined text data and metadata from the tweets to exploit the available data. In addition, we examined the impact of each model on the classification process.

Twelve features that would be straightforward to extract and calculate were selected out of the dataset. These features were extracted from the user account and tweet data, such as the account age, the number of followers, and the number of replies (see Section 5).

The dataset was collected using a premium Twitter API between 1 September 2020 and 1 October 2020, using more than 50 hashtags that were trending during that period in Arabic, according to the website in [9]. The collected data were approximately 1.25 million tweets. Two samples were selected; the first sample (Sample I) was randomly chosen. The second sample (Sample II) was modified after reviewing the data.

The following is a summary of this study:

- Presented an up-to-date survey and analysis about related research to spam detection in Arabic on Twitter.
- Collected a large tweet dataset in Arabic using a premium Twitter API that provided unique features not available to the public. Such a dataset could serve as a benchmark dataset for other researchers in the field.
- Developed a novel framework that combined text-based and metadata-based features for detecting Arabic spam accounts on Twitter using deep-learning algorithms.

- Investigated the effectiveness of combining accessible metadata with textual data when identifying spam accounts on Twitter.
- Benchmarking the proposed framework with the most prominent models of machine learning and deep learning applied to spam detection in Arabic text.

The remainder of this paper proceeds as follows: Section 2 presents related works in spam account detection in Arabic. Section 3 introduces the background of the main algorithm used in this work, namely deep convolution neural networks. Section 4 discusses the methodology used to develop the proposed framework. Section 5 reports the experimental results and discussion, and finally, Section 6 concludes this paper.

2. Related Works

There have been a variety of techniques employed to contain the spread of spam on Twitter. Categorizing these techniques has involved many factors, such as the type of features used, the level of spam detection, and the type of approach used. In this section, studies are classified into machine-learning and deep-learning approaches. The detailed classification was based on that used in [14]. Figure 2 illustrates the Twitter spam-detection classification based on the techniques used to build the detection engine.

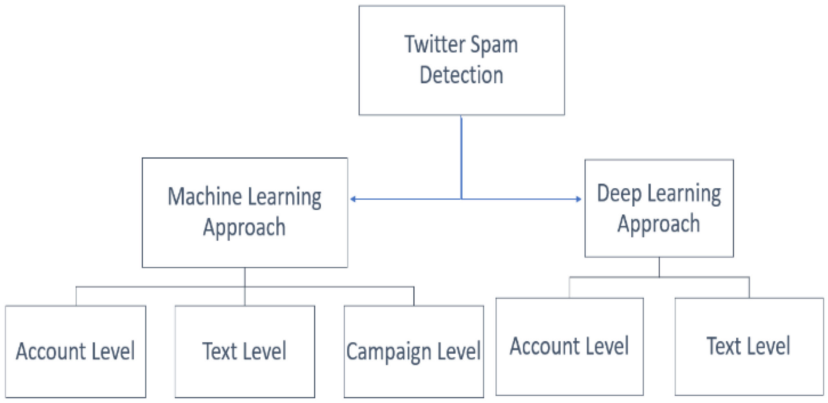

Figure 2. Spam-detection approaches.

2.1. Machine Learning Approaches

Many previous studies have discussed the challenges involved in spam detection in social media, including platforms such as Twitter. These issues increase in complexity as demand and interaction increase on these platforms, which then increases the amount of data to be managed. Therefore, these social platforms have become fertile environments for spam and fraudulent claims via fake ads or impersonations as well as for acquiring private data [15]. One of the most common techniques for detecting spam has been via machine-learning algorithms.

While reviewing the existing studies concerning spam detection on Twitter, we found that many researchers had divided the studies according to the level of detection, and then identified them by the level of detection: tweet, account, or campaign [16].

2.1.1. Account Level

Account level detection relies on extracting features from the user's account, which includes items such as the total number of followers, friends, mentions, etc. Many researchers have used methods that depended on the features at the account level, and they then extrapolated additional features and parameters not previously considered to enhance their machine-learning algorithms [17]. The introduced features yielded better performance than the previous approaches. The researchers in [18] provided further insights regarding the actions of spam accounts on Twitter. They suggested collecting an enhanced set of features that were separate from historical tweets and were only accessible on Twitter for a limited time. A key finding was that automated spam accounts published an average of 12 tweets

per day at specific times. The work of [19] presented a framework that utilized a sample of non-uniform features within a gray-box machine-learning method that utilized different random-forest algorithms to distinguish a spammer among the Twitter traffic.

2.1.2. Tweet Level

Typically, spam accounts send tweets containing inappropriate or unethical content. Therefore, tweet detection methods can track semantics and common meanings. Among the methods used to represent words, term frequency–inverse document frequency (TF–IDF) and bag-of-words have been the most predominant [4].

Many studies have focused on detecting spam using only the content of tweets. The work of [20] detects spam at the tweet level depending on language tools using a support vector machine SVM classifier. The researchers in [21] proposed an unsupervised technique to distinguish legitimate accounts from spam accounts based on tweet contents such as words, tags, and URLs. This was achieved by using a biogeographic optimization algorithm (BGOA). The algorithm executed a chromosome modification on the basis of immigration and emigration rates to achieve a better result. In the study [22], the researchers were able to identify spam by building a model based on a statistical analysis of the language in the tweet without any previous information about the user who posted it.

2.1.3. Campaign Level

Instead of focusing on the characteristics of just one spam account, studies focusing on the campaign level examined the behavior of a group of spam accounts. Usually, these accounts had been directed by a specific party. These directed accounts had common characteristics such as tweet time, number of retweets, and total tweets per day, and this indicated that they are managed by a specific bot or app. Among the studies that focused on this topic was [23], which concentrated in particular on Arabic accounts that were identified as spam. The suggested framework adopted a semi-supervised method that labeled accounts as spam or legitimate, depending on their behavior and account information. The researchers in [24] exploited the URLs in tweets using two methods: The first was similar to that used when detecting spam in email. The second stripped the existing URLs hidden in Google's search results. The tweets involved in this campaign directed unsuspecting users to phishing sites in order to acquire more followers. In the study [25], a multi-topic social spam detection model was presented. The presented method addressed the shortcomings of previous methodologies by performing a fine-grained analysis of user data and metadata using semantic and sentiment analysis, resulting in the extraction of multiple topic-dependent and topic-independent features. Several popular machine-learning algorithms were applied to a labeled Twitter dataset. The Multilayer Feedforward Neural Network or MLFFNN and random forest methods had the best receiver operating characteristic ROC, according to the results. The research in [26] provided a completely unsupervised spammer detection method based on a user's peer acceptability in a social network to identify spam accounts. The peer acceptance of another user was calculated based on the two users' shared interests across various topics. The method did not employ labeled training datasets. Although the approach did not outperform supervised classification-based approaches in terms of accuracy, it proved to be a viable alternative to existing classifiers for spam detection, with 96.9% accuracy.

Table 1 shows a summary of Twitter spam-detection studies using machine-learning techniques. These approaches progressed in the past and had significant results, but the spam accounts soon adapted to some of these approaches. Therefore, researchers have turned to other methods that rely on social graphs to obtain more effective results. The disadvantages of social graphs are their complexity, processing time, and resource demands.

Table 1. A Summary of Twitter spam-detection studies using machine learning.

The Study	Detection Level	The Method	Classifiers	Results
[22]	Tweet-based	Supervised	SVM	F measure: 88.3% False positive: 63% Accuracy: 92.2%
[19]	Account-based Tweet-based	Supervised	RF	The proposed solution illustrates that a non-uniform feature sampling method results in a more reliable predictor compared to traditional approaches.
[24]	Campaign-based	Supervised	Decision-tree regression (DTR)	True positive: 73.5% False positive: 25%
[20]	Tweet-based	Supervised	SVM	Accuracy: 93% True positive: 97.5% False positive: 5%
[17]	Account-based	Supervised	k-NN, DT, NB, RF, LR, SVM, and XGBoost	RF with accuracy: 91% Precision: 92% F1 score: 91
[18]	Account-based	Supervised, suitable for real-time	RF, ET, GB, MLP, MaxEnt, SVM	Many comparisons on different datasets in which Random Forest excels.
[23]	Campaign-based	Semi-supervised	Label propagation and spreading	F measure: 89% Accuracy: 91%
[21]	Tweet-based	Unsupervised	Biogeography Genetic Algorithm	With the 14-user set: Precision: 81% Recall: 1 Accuracy: 85% F measure: 90%
[25]	Account-based	Supervised	Multi-Layer Feed-Forward Neural Network (MLFFNN), GB, GLM, DT, NB, RF	MLFFNN, RFT, and GBT achieve the best ROC means among all other classification models.
[26]	Account-based Tweet-based	Unsupervised	Deep Neural Networks for Bot Detection (DNNBD), Seven Months with the Devils (SMD) and K-mean	Accuracy: 96.9%

2.2. Deep-Learning Approach

Traditional machine-learning methods have little capacity to process raw data without intervention. In recent years, it has taken meticulous engineering and considerable domain knowledge to create a pattern recognition or machine-learning system for the implementation of a feature extractor that converted the raw data to the form of the feature vector, in which a classifier could discover input patterns [27]. As a form of machine learning, deep learning allows computers to learn through data experience. Therefore, it does not require ongoing intervention from a human operator. Through deep learning, the computer recognizes different concepts in a hierarchical manner, which allows it to adapt to complex patterns, which are divided into graphic layers that can then clarify these hierarchies [28].

Spam detection is not a new field of research. It has been under investigation for a decade. While the focus had previously been concerning email and internet spam detection, it has evolved to include social media [29]. With the challenges of machine learning, such as spam-drift and the manipulation of extracted features, deep-learning methods were the logical next step due to their rapid, accurate results without the need for ongoing human support [30].

Several studies discussed the use of deep learning to reduce spam in social networks, specifically on Twitter. To overcome the challenges mentioned above, several studies [30,31] introduced a syntax-based analysis of tweet data using Word2vec extraction. These features were classified using traditional spam recognition algorithms. The work of [32] presented an ensemble approach using five convolutional neural networks (CNNs) for feature extraction from the text as well as one that was based on typical features. Various word-embedding methods were used for the training phase such as Glove and Word2vec. The features varied between content, account, and graph-based features. In addition, the study in [13] presented a new approach based on the principles of deep learning to extract text-based features using Word2vec as well as Bidirectional Long Short-Term Memory or BiLSTM. The deep-learning features were used for training and were tested on different classifiers. The study in [33] used naïve Bayes classifiers and an artificial neural network to predict spam based on text data, but it was limited to the English language. Finally, the authors in [34] proposed a new automatic spam detection method based on a semantic convolutional neural network (SCNN). Word2vec was used to convert textual data and improve the performance of CNNs when used with small-sized data such as tweets. In the study [35], they proposed a model incorporating both graph convolutional networks (GCNs) and Markov random fields (MRFs), which operated on directed social graphs for semi-supervised social spam detection. The experiments were conducted on two real-world Twitter datasets and achieved superior performance. In the study [36], they proposed a GCN-based anti-spam (GAS) model that was a large-scale anti-spam strategy based on graph convolutional networks (GCN) for detecting spam advertising "Xianyu", China's biggest second-hand-products application. A heterogeneous graph and a homogeneous graph were combined to capture the local and global contexts of a comment. Offline and online experiments were conducted and indicated that the recommended strategy was effective. Moreover, a new deep-learning architecture based on convolutional neural networks (CNN) and long short-term memory (LSTM) was developed in [37]. With the use of knowledge bases such as WordNet and ConceptNet, the model was supplemented with semantic information in the representation of words. The use of these knowledge bases enhanced the performance by providing better semantic vector representation of test data that had a random value as they had not been used in training. The researchers in [12] proposed a novel technique for distinguishing faked reviews utilizing linguistic features. To classify the reviews, researchers used unsupervised learning via self-organizing maps (SOM) in association with CNNs. Reviews were converted to images by arranging words with similar semantic meanings around a self-organizing maps (SOM) grid cell. Comprehensive experiments showed that the suggested strategy was effective in both single- and multi-domain scenarios.

Table 2 summarizes approaches relying on deep learning and shows that they use text-based features. They have the ability to discover new information through text, as well as to limit the adaptation of spam accounts by changing the statistical features according to the data. However, the use of lightweight features confirmed and quickly made decisions about accounts' behaviors.

Table 2. A summary of Twitter spam-detection approaches using deep learning.

The Study	Features	Text Extraction Method	Classifiers
[30]	Text-based	Word2Vec	MLP, NB, RF, and DT (C45)
[32]	Text-based, User-based + Graph-based	CNN + Word2Vec	CNN and RF
[31]	Text-based	Word2Vec	MLP
[13]	Text-based	Word2vec + BiLSTM	DT C4.5, and RF KNN, NB, SVM, and NN
[34]	Text-based	SCNN + Word2vec	SCNN
[33]	Text-based	None	NB and ANN
[35]	Graph-based	BoW	GCN and MRF
[36]	Graph-based	PSL + WR	GCN
[37]	Text-based	Word2vec	CNN and LSTM
[12]	Text-based	GloVe 300	SOM–CNN

2.3. Arabic Spam Detection

The studies on Arabic spam detection on Twitter have been limited due to many reasons, the most important of which has been the complexity of the Arabic language, as compared to English, and perhaps the lack of datasets specifically for this purpose. However, according to the statistics, Arab countries are among the most affected by spam campaigns.

The study in [38] used machine-learning algorithms to classify Arabic tweet content of spam accounts. They collected a dataset with almost 1.3 million tweets using the top five Arabic swear words and extracting features at both account and tweet levels. The findings indicated that naïve Bayes (NB) had the best results with an accuracy rate of up to 90%. However, labeling swear words was not sufficient to identify spam activity due to its narrow focus.

The study in [39] concerned Twitter spam accounts, particularly those who tweeted in Arabic, and sabotage trends in Saudi Arabia. Features were obtained from both account profiles and posted content. The study used three machine-learning algorithms, namely, naïve Bayes, random forest, and support-vector machines with a radial basis function (RBF) kernel on Weka. The results showed that RF yielded the best performance, with 92.59% accuracy for the Arabic spam dataset model. However, despite a dataset of more than 23 million Arabic tweets, the authors manually labeled only 5000 tweets as a sample.

The authors in [40] investigated effective techniques of word embedding as text-based features to detect Arabic spam on Twitter. Therefore, the influence of the text data was analyzed to teach the models with word embedding. The evaluation was performed using three machine-learning algorithms: C4.5, SVM, and NB. The word-embedding approaches detected Arabic spam with 87.32% accuracy. However, the study focused on detecting bot spam only and relied on contextual features.

The study in [41] introduced a bot-detection system for Arab spam accounts. The study examined the impact of culture and language on this type of system. It also identified the most important features associated with bot accounts. This system was evaluated using random forest, naïve Bayes, and SVM. The highest accuracy of 98.68% was achieved by the RF techniques.

The authors in [42] focused on Arabic content aimed at adults on Twitter, as they collected tweets with hashtags dedicated to this type of content. They then collected tweets on the same accounts to obtain more information about the characteristics of these accounts. They also analyzed the prevalence of this type of spam according to the geographical location of the account. This model achieved 79% accuracy using the SVM technique.

Using features at the account and content levels, the study in [2] explored spam tweets in the Gulf dialect. The classification model was based on NB and SVM techniques. The NB outperformed SVM in terms of accuracy in detecting spam accounts. Targeting the Gulf dialect was difficult due to the social diversity of the countries, so a more general model that includes more than one dialect of the Arabic language would be more advantageous.

The study conducted in [23] focused on the behavior of groups involved in spam campaigns on Twitter and targeted Arab users. The study used a semi-supervised model to identify spam account features without addressing the content of the tweet, which eliminated potentially useful information. The model achieved results of 89% F measure and 91% accuracy.

In addition, authors in [29] used machine-learning techniques to detect rogue and spam content in Arabic tweets. They gathered a large dataset with 10.1 million tweets from spam accounts posting in Arabic. A total of 47 features from the content and accounts were extracted and analyzed, and the best-performing ones were selected. The results indicated that the random-forest algorithm using 16 pre-tested features achieved an accuracy rate of more than 90%. The drawback of this study was that some spam accounts can appear to be legitimate for a time and post appropriate content.

The study in [43] presented a model aimed at detecting spam at the level of the account or bot in Arabic. The study focused on the content of tweets in terms of "emoji" use, the dialect used, and the hashtags with URLs. Using AraBERT, which is a BERT-based model

that was trained on Arabic news articles along with Arabic Wikipedia and SVMs, the model achieved an F1 score of more than 98%.

What limited the previous studies in Arabic spam detection was that most were trained for the dialect of one country (e.g., Saudi, Egyptian, Iraqi, etc.). Moreover, most studies extracted features using only one type of data, either text or binary, which overlooked important information. In addition, most studies relied on labeling data in only two ways, either manually or using keywords. Both methods are ineffective and subject to personal biases, and spam accounts easily bypass them.

The use of deep-learning approaches, instead of machine learning, has several advantages, including more accurate results and improved speed, as extracting features does not require manual work or high computational resources.

In terms of the availability of datasets of Arabic Twitter spam, to the best of our knowledge, there was no suitable one that existed for our study. Most studies have focused on sentimental analysis, corrupted and polluted content, or very small datasets. Therefore, the researchers involved in this study collected a large Arabic dataset for this research that may also be used by other researchers in the future. Table 3 summarizes existing Arabic Twitter spam-detection studies.

Table 3. A summary of Arabic Twitter spam-detection studies.

Study	Approach	Detecting Level	Features	Labeling	Classifiers
[7]	Machine learning	Account-level	Content-based	–	Support Vector Machine (SVM)
[39]	Machine learning	Account-level	Content-based + Account-based	Manual	Naïve Bayes, Random Forests, and SVM
[40]	Machine learning	Tweet-level	Content-based	Manual	Naïve Bayes, Decision- trees, and SVM
[42]	Machine learning	Tweet-level	Content-based	Keywords	SVM
[41]	Machine learning	Account-level	Account-based	Manual	Random Forest, Naïve Bayes, and SVM
[2]	Machine learning	Tweet-level	Content-based + Account-based	Keywords	Naïve Bayes and SVM
[23]	Machine learning (semi-supervised)	Campaign- level	Content-based + Account-based	Manual	Label propagation and spreading
[36]	Machine learning	Tweet-level	Content-based + Account-based	Keywords	Random forest
[43]	Machine learning	Tweet-level	Content-based + Account-based	Manual	SVM

3. Convolutional Neural Networks

A convolutional neural network (CNN) is a neural network with several layers that are sparsely connected and designed for complex data features. When using a CNN with text data, converting data into numerical values should be carefully considered. Therefore, embedding is typically used at this stage to convert texts into word vectors by creating a two-dimensional matrix corresponding to the sentences. Therefore, each of the rows contained in this matrix conforms to a specific word or token [34]. CNNs are known for their ability to recognize and process images, support robot vision, and their use in autonomous vehicles. The CNN algorithm consists of several stages: the convolutional layer, the pooling layer, and the fully connected layers. Figure 3 illustrates CNN layers which are explained in the following subsections as well.

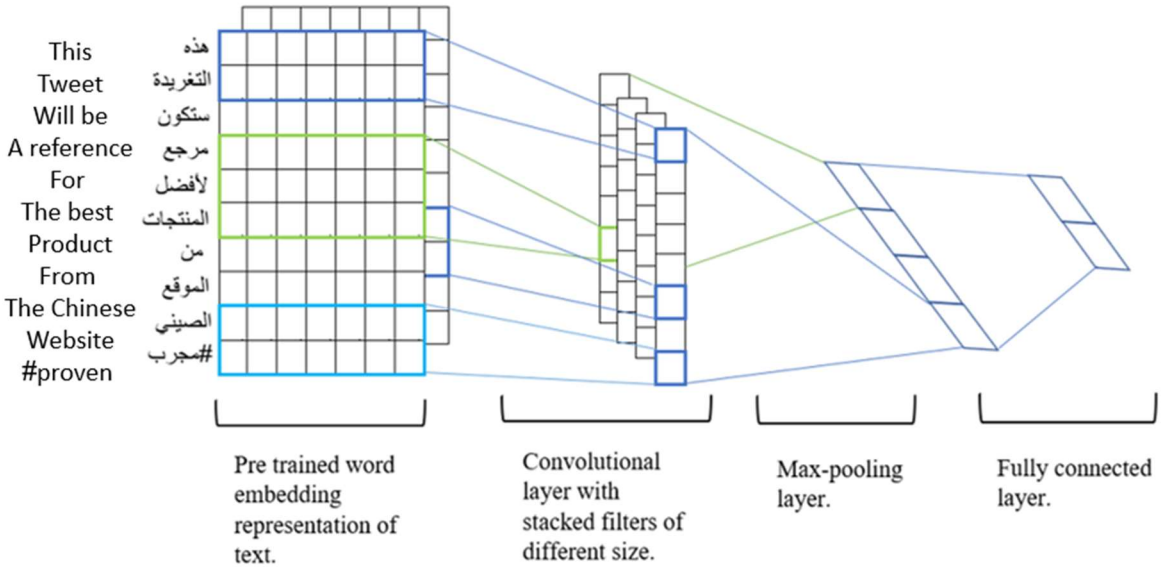

Figure 3. Architecture of CNNs.

3.1. Convolution Layer

The algorithm is based on the principle of the "sliding window" when used with number matrices. At this stage, convolution layers are constructed using vector words. The array filters features by data type to produce a feature map. Next, the values are multiplied by the filter element-wise with the subset of the input matrix, and they are combined. The matrix resulting from this stage is equal to the filter matrix and is called a convolved matrix [34]. Figure 3 shows the sliding-window technique used in the convolutional layer, where the filter slides in one direction using two words from the sentence simultaneously. It calculates an element-wise output of the weight of every word and is then multiplied by the weights assigned to the convolutional filter.

3.2. Pooling Layer

The main purpose of this layer is to standardize the output produced from the convolutional layer to obtain inputs that can be fed into the classification layer. In addition, the pooling layer reduces the dimensions generated from the previous layer while preserving the integrity of important features and bypassing the obstacle to overfitting. Next comes the maximum function, which is the most important element of this layer. It is based on taking the maximum value of the cell that resulted in the window of cells. To obtain a univariate vector, the resulting matrices from all filters are then connected [34]. Figure 4 illustrates the manner in which the pooling layer works.

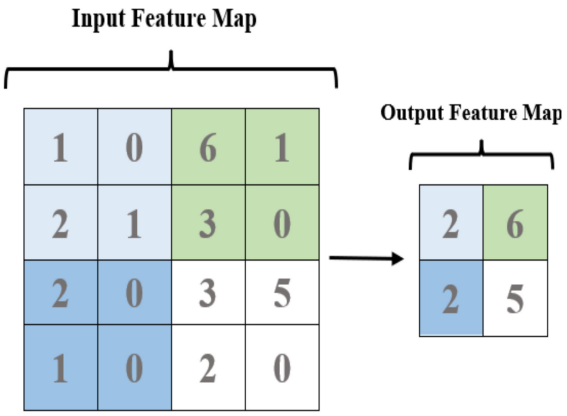

Figure 4. Pooling layer.

3.3. Fully Connected Layer

The fully connected layer (FC) is the conclusion of the architecture of the CNN algorithm. This layer receives the output from the pooling layer transformed as a one-dimensional feature vector. This process is called flattening. Figure 5 illustrates this process whereby the output of this layer is a vector representing the number of expected classes, such as, for example, spam or not spam [44].

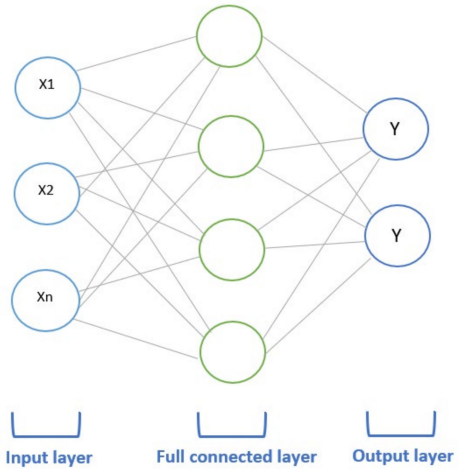

Figure 5. Fully connected layer.

4. Proposed Framework

As mentioned in Section 1, the proposed framework constituted two separate models, a text-based model and a combined model, both of which used CNNs and considered various feature sets. Figure 6 shows the abstract architecture of the framework and its components.

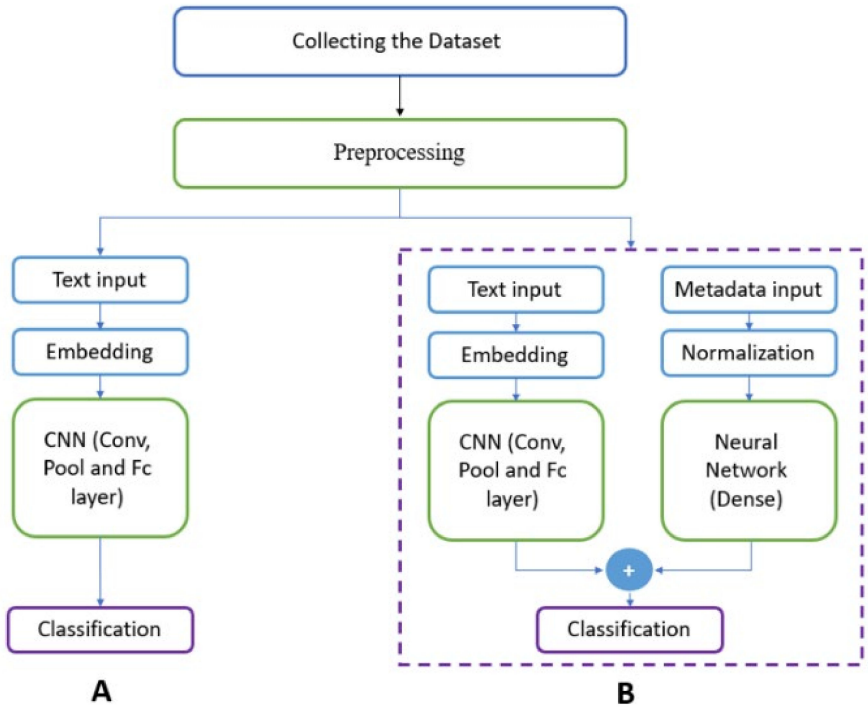

Figure 6. The proposed framework: (**A**) text-based model and (**B**) combined model.

4.1. Text-Based Model

This model had three stages: embedding, CNN algorithm, and classifications.

Embedding each word in a tweet converted them before passing them on to the CNN algorithm. Every word was presented as a high-dimensional vector, as shown in Figure 7. The selection, between 128-dimensional and 200-dimensional word-embedding to discover what worked best with Arabic text and produced a better result, was considered. According to studies [31], 200-dimensional Word2Vec was the best option that represented each word with a vector of 200 dimensions. Word2Vec uses two models to create an embedding. The continuous bag of words (CBOW) and skip-gram are the best models to use with the Arabic language. As illustrated in Figure 7, CBOW embeds a word by predicting the word itself given its neighboring words while skip-gram embeds a word by predicting the words that proceed or precede it. The rich nature of the Arabic language contributes directly to the influence of a corpus on creating accurate word-embedding according to [45], which used a self-made corpus from 16,000 tweets.

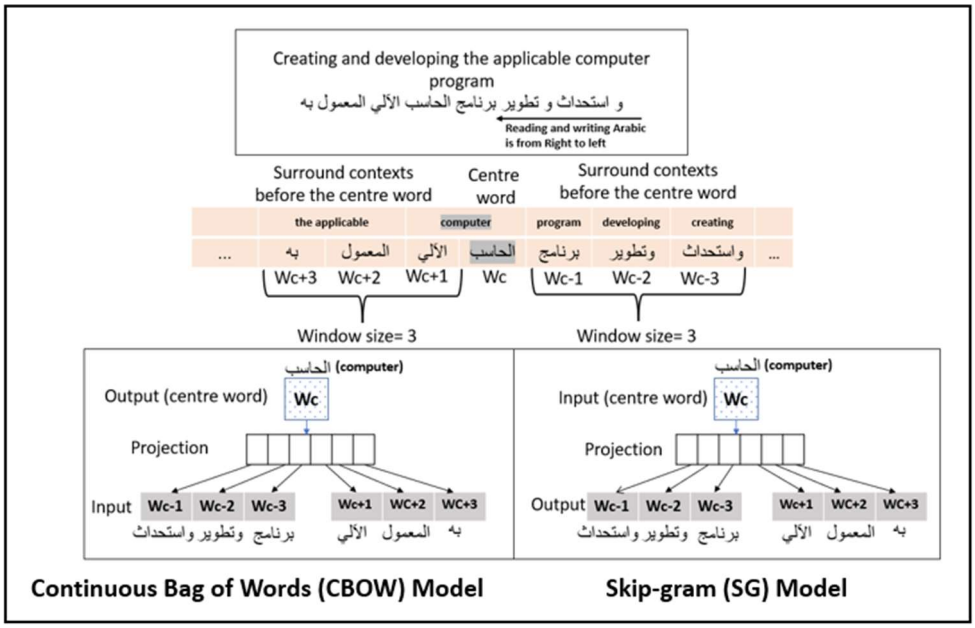

Figure 7. A model for Arabic Word2Vec [45].

The output tweet was a matrix that had a different length based on the number of words each tweet contains; therefore, a padding technique was required. The padding technique was used to ensure that all tweet matrices shared the same length. Subsequently, we determined the maximum length of the tweets; that is, a tweet that did not reach the maximum length would be padded with a value of zero. Once the embedding step was completed, a high-dimensional matrix was sent to the CNN algorithm.

As previously illustrated in Section III, a CNN algorithm is made up of "neurons" with learnable weights and biases. It contains three layers. The convolution layer (CONV) is the building block of the CNN and its main goal is to extract features from the input data. The pooling layer (POOL) applies a function for reducing the input dimensionality. The fully connected layer (FC) is used to generate the output vector, which has the same dimensions as the number of potential output classes, spam, and not spam. After reviewing many scientific papers that used CNNs with Arabic text data, different building structures were tested [46,47]. Finally, we chose a simpler architecture with an embedding layer with 200 dimensions as it provided better results when working with text, according to [44], with a one-dimensional convolution layer (Conv1D) with 128 neurons and a rectified linear unit (ReLU) as the activation function. Next, one-dimensional maximum pooling was applied, followed by a dropout at 5.0. Finally, a fully connected layer with the prediction was obtained.

During classification, the SoftMax function was applied to predict the class result. SoftMax is a well-known function for estimating event probabilities. This experiment had two potential outputs (i.e., spam and not spam).

4.2. Combined Model

To investigate whether an account was spam or not required extracting data from both the account and the posts. We decided to use a model that combined the output provided by the text-based model (text model) with a metadata model using the features extracted in the preprocessing stage, which are presented in Table 4. The features were divided into standard and premium features. Standard features were obtained by a free standard account that had limited accessibility to account and tweet characteristics. Premium features required a paid subscription to access all the existing features presented by Twitter's API. In this model, a neural network was used with five fully connected layers, based on [44]. The first stage normalized the data and then sent them to a neural network, as explained below.

Table 4. Premium features vs. standard features.

Standard Features	Premium Features
Number of followers.	Number of replies.
Number of followings/friends.	Number of retweets.
Number of favorites.	Account verification.
Number of tweets.	
Number of lists.	
Number of characters per tweet.	
Friends and followers ratio.	
Account reputation.	
Account age.	

The features were normalized using Sklearn's MinMaxScaler before sending them to a neural network. Based on the literature, normalization was an important step before sending numeric data to a deep-learning algorithm. It would improve overall performance including faster learning and higher accuracy. The outputs from this phase were features that were similar in scale, with a mean between 0 and 1.

Five fully connected neural network layers (Dense) were used in the "dense" step. The layers were 512, 256, 128, 64, and 32 for a better result, as suggested by [44]. In addition, an extra layer was used to guarantee that the output of the metadata model had the same dimensions as that of the text-based model.

The concatenation and classification step applied a fully connected layer using the SoftMax activation function to classify the concatenate text and metadata model outputs. The final prediction was provided, which specified whether the account was a spam or not spam account.

5. Experimental Results

The dataset contained 1.25 million Arabic tweets, and it was used to answer the following questions:

- How effective were the combined model, as compared to the existing machine-learning and deep-learning models?
- How effective is the suspended account method for labeling an Arabic dataset?
- Will the premium features have effects on the model performance?

To answer these aforementioned questions, two dataset samples (Sample I and Sample II) were selected after the necessary data preprocessing. The performance of the proposed framework was then evaluated in terms of accuracy, precision, recall, and F1 score, and then analysis and discussion of the findings were reported.

5.1. Collecting Dataset

It was difficult to find an Arabic dataset that was suitable for spam detection. Most of the available datasets were either too small, private or otherwise incompatible with this research. Therefore, we collected our own dataset. These data were subject to the new Twitter regulations that have blocked many features that were previously available. Moreover, it was necessary to subscribe to the Twitter premium service to obtain all available features. Twitter premium is a subscription service with a monthly fee and is divided into categories according to the needs of the user, researcher, and even commercial companies so they can benefit from the data.

The data were obtained from Twitter by collecting tweets through trending hashtags. The Statista.com website was used to obtain the top 50 interactive Arabic hashtags at the time of the study, and then the tweets were downloaded using Twitter's API. The data were downloaded for a whole month between 9 September 2020 and 9 October 2020. A total of 1.2 million tweets were divided into six files at a rate of 250,000 tweets per file and then saved to a file with a JSON extension.

After estimating the features, the files were extracted into a CSV file, so that the necessary processing operations could be performed. The data were divided into two types: text data, which was the text contained in the tweet, and metadata, which were mostly numeric and were extracted directly or through some calculations.

5.2. Labeling the Dataset

Labeling the data was simple but arduous, and a large number of accounts increased the time requirements. Approximately 16,700 users were considered, and their tweets were constructed and labeled. Labeling was carried out with Postman, which is an application used for API testing. It is an HTTP client that tests HTTP requests, and through which we obtained different types of responses that needed to be subsequently validated. When creating an app within a project on the Twitter developer platform, it obtains permissions to search for users and verify their accounts. In this study, (https://api.twitter.com/1.1/users/lookup.json, accessed on 21 March 2021) the end-point was used, which allowed up to 100 users per search by user handle or account ID, which was a unique set of numbers assigned to every account on Twitter. If a user profile and their latest tweets were found, the account existed; if it did not, they may have been suspended from Twitter or deleted their account. As mentioned, two samples were selected from the labeled dataset.

Sample I was randomly selected from the labeled dataset, whereas Sample II was manipulated upon observing the data content of the tweets. For example, some accounts were determined to be spam, but their data features indicated otherwise (e.g., long-established account, etc.). Other values that were flagged included the ratio of friends to followers, the account overall age, and the number of tweets and interactions. These indicated that these accounts may be legitimate accounts, but recent updates to Twitter's rules may have suspended the accounts due to accidental or incidental violations that were not within the actual intention of the regulations. This corresponded to the lack of awareness about these new regulations among Arab users and even those among them who were highly influential and active on Twitter. These included possible infringements on copyright and intellectual property rights as well as commentary regarding religions and sects. These accounts may undergo long suspensions due to the difficulties involved in account recovery, such as language barriers with technical support, expensive mediators who specialize in recovering social media accounts, etc. Some users abandon their accounts and create new ones or permanently delete their accounts. After reviewing and verifying the data, some accounts were re-labeled accordingly; then, the Sample II dataset was prepared. After examining the results of Sample II, we noticed a significant improvement in the model's performance. However, better results could be obtained by reducing the ratio between legitimate and spam accounts, especially to avoid imbalanced data. Figure 8 shows the number of accounts per dataset sample.

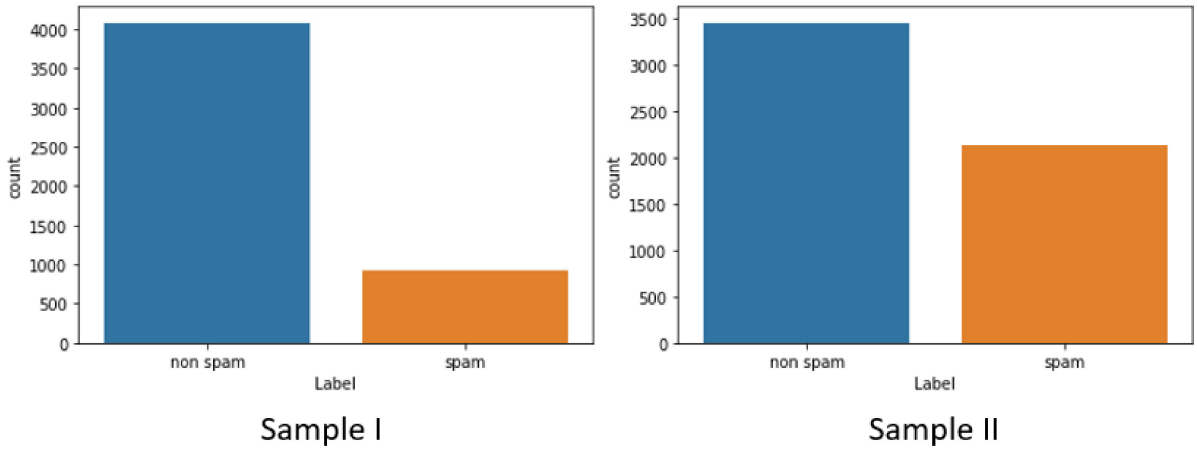

Figure 8. Number of users per sample.

5.3. Pre-Processing

At this stage, the preprocessing operations on the two types of data are presented separately.

5.3.1. Text Data

The text data had to be cleaned as each tweet may contain excessive data that would not benefit our research and therefore result in unnecessarily slow processing. The existing literature provided cleaning guidelines that were applied to the text data including:

- Repeating letters;
- Formation (Tashkeel);
- Emoji;
- Punctuation marks;
- White spaces;
- Numbers;
- Words in languages other than Arabic.

The cleaning method used in this paper was adopted from [48].

5.3.2. Metadata

After reviewing related research, 12 reliable features were selected. The features needed to be streamlined and could not require complex operations for extraction. These characteristics focused on the account details, the user behaviors, and the tweet content. The following is a detailed explanation of these characteristics, as stated by Twitter policy [49].

(A) Graph-based features

Ratio of friends to followers

To calculate this feature, the number of friends was divided by the number of followers.

$$\frac{Number\ of\ followers}{Number\ of\ friends}$$

If the ratio result was small, then the likelihood that the account was spam increased [49].

Account Reputation

This feature was obtained by dividing the number of followers by the number of followers plus the number of friends.

$$\frac{Number\ of\ followers}{Number\ of\ friends + Number\ of\ followers}$$

If the result was very small and near zero, the account was likely spam, as spam accounts tend to gain more followers [49].

Account Age

Twitter spam accounts are often quite young, and this is mostly due to their discovery and prevention by Twitter, as these spam users must resort to creating new accounts [15,50].

$$\frac{\sum(Tweet\ create\ time - User\ create\ time)}{Tottal\ tweets\ number}$$

(B) Account-based features

Number of followers
Indicates the number of other accounts that follow the subject account.
Number of followings/friends
Indicates the number of other accounts the subject account is following.
Number of favorites
Indicates the total number of tweets that have been "favorited" by an account.
Number of tweets
Indicates the total number of tweets by an account up until the date the data were extracted.
Number of replies
Indicates the total number of responses from the account to other accounts.
Number of lists
Indicates the number of lists where the account has been added.
Number of retweets
Indicates the total number of retweets the account has published.
Account verification
Indicates if the account has been verified by Twitter or not.

(C) Tweet content features

Number of characters per tweet
Indicates the number of characters a tweet contains.

5.4. Simulation Environment

The Keras functional API was used in this study to combine the two models. Keras is a tool for deep learning that utilizes TensorFlow as a back-end for the implementation of deep-learning models. The experimental models were run on a Windows 10 operating system with an Intel Core i5 processor and 10 GB of RAM.

5.5. Parameters' Settings

The models were trained by Adam optimizer with 100 epochs for the text-based models and 50 epochs for the combined model.

5.6. Performance Matrix

Four known standard metrics were used: accuracy, precision, recall, and F1 score. Accuracy is the ratio of the total number of instances that are classified correctly for both classes over the total number of instances. Precision is defined as the rate of the number of correctly classified instances (true positives) to the total number of instances (true positives and false positives). Recall refers to the ratio of the number of instances correctly classified (true positives) to the total number of predicted instances (true positives and false negatives). The F1 score (F1) is the average of precision and recall, calculated as the F Measure = (2 × Precision × Recall)/(Precision + Recall).

5.7. Framework Evaluation

The proposed framework was evaluated by comparing the combined model and text-based model to a simple metadata model. Then, a comparison with existing machine-

learning and deep-learning models was performed. A comparison between the combined model with standard and premium features is presented.

5.7.1. Evaluating the Combined Model

The first step in evaluating the combined model was to compare the performance in terms of accuracy and loss. Loss is a number indicating how poorly the model's prediction was for a single sample. If the model's prediction was perfect, the loss was zero; otherwise, the loss was greater. The comparison was conducted under a number of different epochs such as 20, 50, and 100. As shown in Table 5, the model performed well at 50 and 100 iterations for Sample I, with a very small difference between them, while the performance dropped slightly at 20 iterations. On the contrary, the model in Sample II had better results at 20 and 100 iterations while it regressed slightly at 50 iterations. Furthermore, Figure 9 illustrates the model's accuracy and loss for Sample I. The model appeared to learn poorly from the existing sample, and it was underfitted, as demonstrated in Figure 9. However, the performance had improved, producing a better result at 100 iterations. This indicated that the sample could not properly train the model. Moreover, the model's performance clearly improved significantly in Figure 10 with Sample II, overcoming the previous issue. The model curve in both accuracy and loss evolved after every increase in epochs, showing a model with a relevant fit in training and testing.

Table 5. Combined model evaluation.

Number of Epochs	Sample I		Sample II	
	Accuracy	Loss	Accuracy	Loss
20	81.81	44.57	93.11	16.87
50	82.12	43.96	92.66	16.80
100	82.01	44.48	94.27	15.48

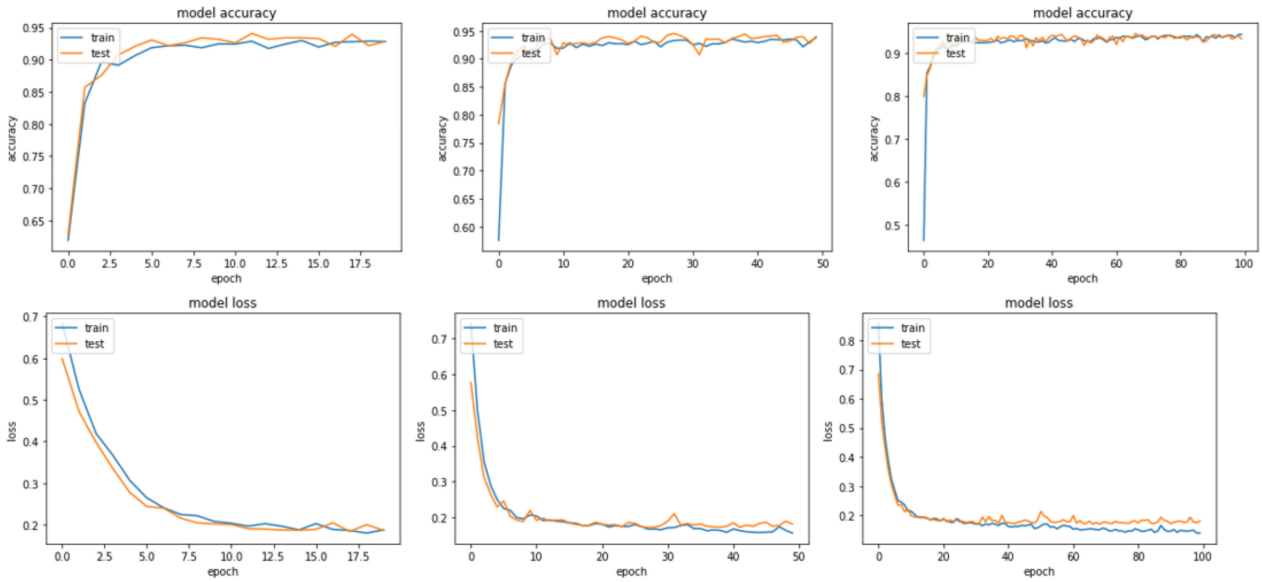

Figure 9. Sample I accuracy and loss curves for epochs (20, 50, 100).

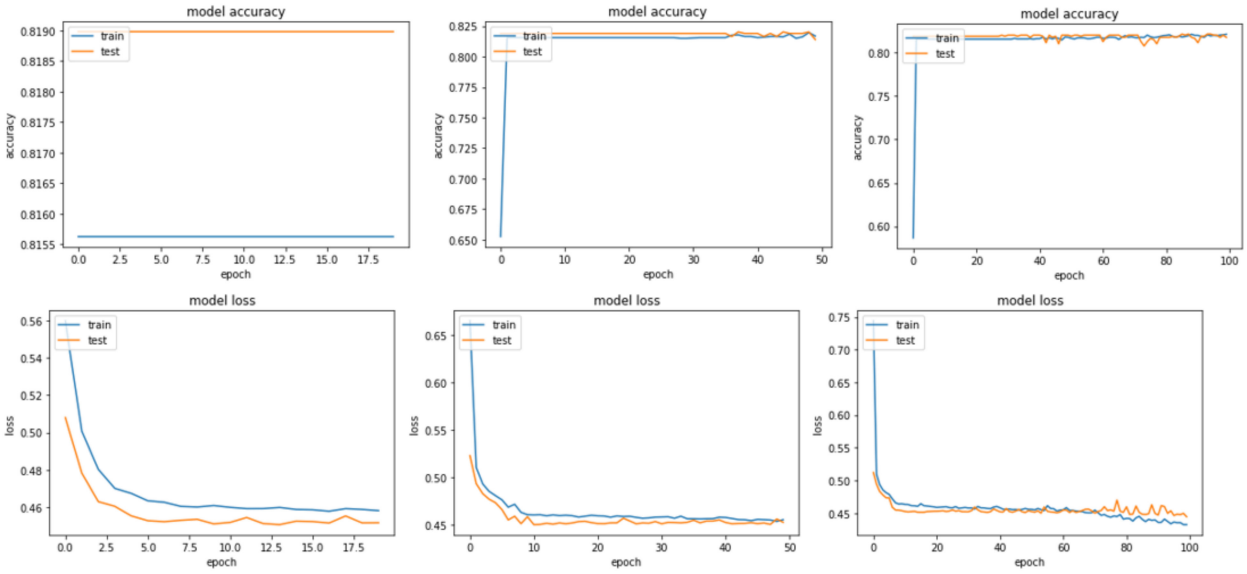

Figure 10. Sample II accuracy and loss curves for epochs (20, 50, 100).

5.7.2. Combined, Metadata, and Text-Based Model Comparison

Most of the models in the literature have depended on features extracted from the user account (e.g., number of tweets, number of followers, etc.) or the content of the tweets (e.g., number of characters, number of words, etc.). To our knowledge, no one has combined the metadata with the tweet text to detect spam in Arabic content.

Therefore, to demonstrate the effectiveness of the proposed framework for detecting Arabic spam on Twitter and the impact of combining data, a comparison between the combined model with a CNN text model and a simpler model that relied only on metadata was conducted. Table 5 shows the detection performance of the three models using Sample I and Sample II.

As shown in Table 6, in terms of accuracy and recall, the combined model using Sample I outperformed the other two models with 82.02% accuracy, as compared to 82% and 80% for the metadata model and the text-based model, respectively. In terms of precision, the three models obtained similar results, with the metadata model outperforming them at 77.9%. Unsurprisingly, the text model outperformed the other models with a 78% F1 score, surpassing the metadata model by 74.9% and the combined model by 75.31%. This was due to the fact that the F1 score was influenced by precision and recall. However, it would behave differently when lower numbers were present, as it would attribute a higher weight to them [51].

Table 6. Combined, text, and metadata model comparison.

	Sample I				**Sample II**			
	Accuracy	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Text-based Model	80	77	80	78	66.3	65.5	66.2	65.8
Metadata Model	82	77.9	82	74.9	86.1	86.4	86.1	85.8
Combined Model	82.02	77.68	82.02	75.31	94.27	94.33	94.27	94.23

The overall results of the text and metadata models were less than expected. Since we could identify one major issue where some legitimate accounts were errantly marked as spam, we reviewed the labeling of Sample II and manually checked the accounts for auditing; after this, we obtained better results. The combined model using Sample II obtained a much better result, as compared to the previous results using Sample I. Consequently, this was accompanied by a decrease in the results of the text model. A

possible explanation for this could be related to the diversity of Arabic dialects in the data sample, which made it more difficult for the model to identify and connect words.

In terms of accuracy, the combined model outperformed the metadata model at 94.27%, as compared to 86.1%, and the text model at 66.3%. In addition, for precision, the combined model scored the best with 94.33%. Precision was an important evaluation measure in this research as it calculated the false positives and was the number of correctly predicted spam accounts out of the overall predicted spam accounts [52]. The combined model outperformed the metadata model in a recall, similar to their results for accuracy, except that the text model regressed slightly at 66.2%. In terms of their F1 scores, the combined model obtained the best score among the three models with a score of 94.23%.

Spam accounts deliberately use colloquial semantics in their tweets as this increases the difficulty in recognizing them [43]. Overcoming such obstacles required training the script model on a larger and more comprehensive corpus that focused on social media data with respect to various dialects. The metadata model obtained 86.1% with sample II, proving the advantage of the proposed combined framework by using tweet text data.

5.7.3. Comparison with Existing ML and DL Models

As revealed in Table 6, the combined model obtained the best results among the experimental models. Therefore, we conducted more analyses to compare the results of the proposed models with existing machine-learning and deep-learning models.

Tables 7 and 8 reports a summary of the accuracy, precision, recall, and F1 scores for the chosen machine-learning and deep-learning models versus the proposed framework. Four machine-learning algorithms were used: SVM, decision-tree (DT), NB, and logistic regression (LR), and one deep-learning model, namely, LSTM. All models used the two sample datasets.

Table 7. Summary of Sample I results.

			Accuracy	Precision	Recall	F1 Score
ML	SVM		79.82	63.71	79.82	70.86
	DT		71.53	71.26	71.53	71.39
	NB		36.36	75.24	36.36	36.66
	LR		79.82	63.71	79.82	70.86
DL	LSTM	Text Model	79.7	76.6	79.7	77.6
		Combined Model	81.7	76.5	81.7	76
	The Proposed Framework	Text Model	80	77	80	78
		Combined Model	82.02	77.68	82.02	75.31

Table 8. Summary of Sample II results.

			Accuracy	Precision	Recall	F1 Score
ML	SVM		76.10	77.52	76.10	76.45
	DT		91.05	91.11	91.05	91.07
	NB		62.40	77.83	62.40	61.51
	LR		85.41	85.93	85.41	85.11
DL	LSTM	Text Model	63.7	64.1	63.7	63.9
		Combined Model	93	94	93.8	93
	The proposed Framework	Text Model	66.2	65.5	66.2	65.8
		Combined Model	94.27	94.33	94.27	94.23

The results of Sample I show the superiority of the proposed framework in terms of accuracy with a percentage of 82.02% for the combined model, followed by LSTM with 81.7%, which also confirmed the superiority of deep-learning combined models in terms of accuracy. The text-based model achieved 80%, surpassing SVM and LR at 79.82% each. The LSTM scored good results with the combined model at 81.7% while the text model achieved a score of 79.7%. In terms of precision, our models were also superior with a combined score of 77.68% and the text model at 77%. SVM and LR did not perform well

in terms of precision, as they each achieved 63.71%. DT and NB scored a better result in precision with 71.26% and 75.24%, respectively. LSTM obtained a closer result, as compared to both models, with 76.6% by the combined model and 76.5% by the text model. The results of recall also showed that our combined model outperformed the others by 82.02%, followed by the LSTM-combined model with 81.7%, SVM and LR with 79.82% each, while NB performed poorly with 36.36%. Furthermore, our text model scored marginally higher at 80%, as compared to the LSTM text model with 79.7%. In addition, our text model advanced in terms of the F1 score with 78%. Next was the LSTM-combined and text models with 77.6% and 76%, respectively. The combined model, surprisingly, did not perform as expected, as compared to the LSTM model; however, it outperformed both DT with 71.39% as well as the SVM- and LR-based models at 70.86%.

Regarding the results of Sample II, our framework showed significant development in the results of the combined model, as it achieved a score of 94.27% in accuracy. LSTM scored a slightly similar result for the combined model with a total of 93%. DT and LR achieved good results, with 91.05% and 85.41%, respectively. SVM preceded NB with 76.10%. NB continued to do poorly in accuracy with only 62.40%; however, the results were improved, as compared to the previous sample. However, though our text model underperformed in this analysis, it still outperformed the LSTM text model, 66.2% to 63.7%, respectively. In terms of precision and recall, our combined and text models exceeded the LSTM models by 94.33%, 94.27%, 65.5%, and 66.2% respectively. The LSTM-combined model achieved 94% and 93.8% while DT followed with 91.11% precision and 91.05% recall, surpassing LR and SVM. NB compensated for its delay in precision at 77.83%, as compared to the ML algorithms. Our framework achieved the highest results in terms of the F1 score, obtaining 94.23% and 65.8%.

Finally, looking at the above results, it appeared that our framework, and specifically the combined model, was more effective in detecting spam accounts on Twitter. According to [53], SVM has been the most suitable model for Arabic text analysis, as compared to machine-learning techniques. Naïve Bayes can also be used for obtaining a high level of accuracy for a high dimensionality of inputs. The novelty of our presented framework was the result of combining the same features used in ML algorithms with more complex features from the text contents of tweets to indicate account legitimacy. The results of existing machine-learning models have been declining due to the increase in imbalanced data. Real-world data from social networks contain a much lower percentage of spam data, as compared to legitimate data.

5.7.4. Premium Features vs. Standard Features

In this experiment, a simple comparison using the combined model was made between the premium features dataset, including retweet count, reply count, and favorite count, vs. the standard features dataset. Previously, the results showed that Sample I did not perform well and that the model could not learn enough from the dataset in this sample, especially in short training periods, due to the aforementioned issues involving Twitter's changing regulations and accidental account suspensions. Once again, Sample I's data were presented to the model as extra noise that did provide a benefit. After this problem was corrected in Sample II, we observed that the model could benefit from the premium feature data; Table 9 shows the effect. A 10-fold cross-validation was applied to this model, as shown in Figures 11 and 12.

Table 9. Premium features vs. standard features.

	Sample I	**Sample II**
	Accuracy	
Premium Features	82.02	94.27
Standard Features	82.22	93.73

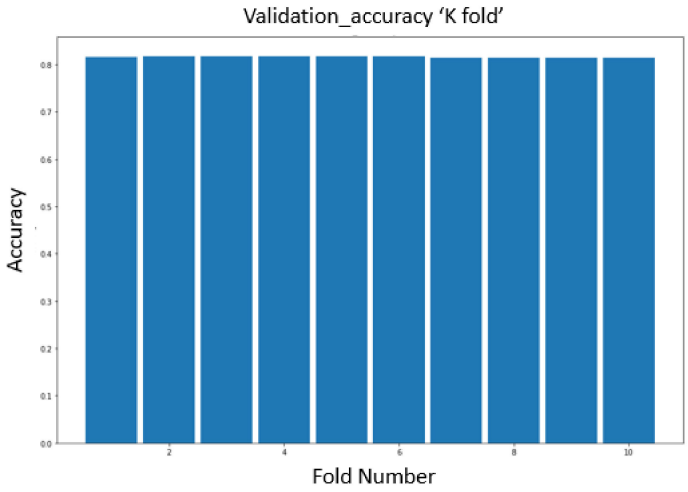

Figure 11. 10-fold cross-validation for Sample I.

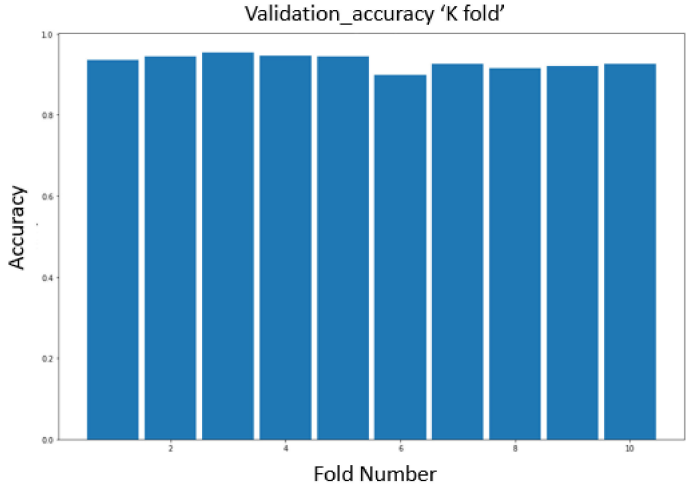

Figure 12. 10-fold cross-validation for Sample II.

6. Conclusions

The purpose of this study was to determine whether a combined framework of text and metadata could be effective for improving spam detection of Arabic Twitter accounts. Furthermore, this study investigated whether account suspensions were indicative of Arabic spam accounts. To verify this, data were collected using Twitter's premium API, which offered features not found in standard data collection. The results showed the superiority of our framework, as it achieved the best results in accuracy in the combined model at 94.27%. The text-based model using CNN performed well with 80% accuracy, despite the difficulties presented with tweets in Arabic and its high sensitivity. Many Arabic dialects and colloquial phrases overlap in communication via social networking sites such as Twitter. This complicates detecting spam accounts using only text-based features and requires many prior steps to obtain accurate classifications. A preprocessing step that could address Arabic dialects with minimal effects on the intention and semantics would be a useful area for further work.

A comparison between the use of Twitter's premium features vs. the standard features was conducted, and this also showed the superiority of using these features to enhance the performance of spam detection.

In addition, this research raised an important question about whether account suspensions were indicative of Arabic spam accounts. The dataset was collected shortly after Twitter implemented its new regulations regarding copyright and intellectual property

infringements. Therefore, many legitimate Arab users unintentionally violated these rules. Therefore, account suspensions were not as useful as a marker for spam detection as initially expected. Collecting a new data sample after the application of these new regulations has settled would allow future researchers to observe the changes in the behavior of users and to reevaluate the usefulness of account suspensions as a parameter.

In the future, we intend to further examine the classification of Arabic dialects used on social networks and to extend the application of our framework to other popular OSNs such as Facebook and Instagram.

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

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Acknowledgments: The researchers would like to thank the Deanship of Scientific Research, Qassim University, for funding the publication of this project.

Conflicts of Interest: The authors declare that they have no conflict of interest to report regarding the present study.

References

1. Sun, N.; Lin, G.; Qiu, J.; Rimba, P. Near real-time twitter spam detection with machine learning techniques. *Int. J. Comput. Appl.* **2020**, 1–11. [CrossRef]
2. Alorini, D.; Rawat, D.B. Automatic spam detection on gulf dialectical Arabic Tweets. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019.
3. Antonakaki, D.; Fragopoulou, P.; Ioannidis, S. A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks. *Expert Syst. Appl.* **2021**, *164*, 114006. [CrossRef]
4. 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]
5. Güngör, K.N.; Erdem, O.A.; Doğru, İ.A. Tweet and Account Based Spam Detection on Twitter. In Proceedings of the International Conference on Artificial Intelligence and Applied Mathematics in Engineering, Antalya, Turkey, 20–22 April 2019.
6. AlKhwiter, W.; Al-Twairesh, N. Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM. *Comput. Speech Lang.* **2020**, *65*, 101138. [CrossRef]
7. Abozinadah, E.A.; Jones, J. Improved micro-blog classification for detecting abusive Arabic Twitter accounts. *Int. J. Data Min. Knowl. Manag. Process (IJDKP)* **2016**, *6*, 17–28. [CrossRef]
8. Wei, F.; Nguyen, U.T. Twitter Bot Detection Using Bidirectional Long Short-term Memory Neural Networks and Word Embeddings. In Proceedings of the 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, 12–14 December 2019.
9. Statista. Twitter-Statistics & Facts. 2020. Available online: https://www.statista.com/topics/737/twitter/ (accessed on 25 January 2022).
10. Elzayady, H.; Badran, K.M.; Salama, G.I. Arabic Opinion Mining Using Combined CNN-LSTM Models. *Int. J. Intell. Syst. Appl.* **2020**, *12*, 25–36. [CrossRef]
11. Alshammari, R. Arabic text categorization using machine learning approaches. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 226–230. [CrossRef]
12. Neisari, A.; Rueda, L.; Saad, S. Spam review detection using self-organizing maps and convolutional neural networks. *Comput. Secur.* **2021**, *106*, 102274. [CrossRef]
13. Ban, X.; Chen, C.; Liu, S.; Wang, Y.; Zhang, J. Deep-learnt features for Twitter spam detection. In Proceedings of the 2018 International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec), Santa Clara, CA, USA, 10–11 December 2018.
14. Washha, M. *Information Quality in Online Social Media and Big Data Collection: An Example of Twitter Spam Detection*; Université de Toulouse, Université Toulouse III-Paul Sabatier: Toulouse, France, 2018.
15. Herzallah, W.; Faris, H.; Adwan, O. Feature engineering for detecting spammers on twitter: Modelling and analysis. *J. Inf. Sci.* **2018**, *44*, 230–247. [CrossRef]

16. Washha, M.; Qaroush, A.; Mezghani, M.; Sedes, F. Unsupervised collective-based framework for dynamic retraining of supervised real-time spam tweets detection model. *Expert Syst. Appl.* **2019**, *135*, 129–152. [CrossRef]
17. Alom, Z.; Carminati, B.; Ferrari, E. Detecting spam accounts on twitter. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 28–31 August 2018.
18. Inuwa-Dutse, I.; Liptrott, M.; Korkontzelos, I. Detection of spam-posting accounts on Twitter. *Neurocomputing* **2018**, *315*, 496–511. [CrossRef]
19. Meda, C.; Ragusa, E.; Gianoglio, C.; Zunino, R.; Ottaviano, A.; Scillia, E.; Surlinelli, R. Spam detection of Twitter traffic: A framework based on random forests and non-uniform feature sampling. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, CA, USA, 18–21 August 2016.
20. Gharge, S.; Chavan, M. An integrated approach for malicious tweets detection using NLP. In Proceedings of the 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 10–11 March 2017.
21. Vishwakarma, R.; Gautam, P.G.P. Biogeography Genetic Algorithm Based Social Platform Spammer Identification Using Content Feature. *Int. J. Eng. Trends Technol. (IJETT)* **2020**, *68*, 19.
22. Martinez-Romo, J.; Araujo, L. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Syst. Appl.* **2013**, *40*, 2992–3000. [CrossRef]
23. Alharthi, R.; Alhothali, A.; Moria, K. Detecting and Characterizing Arab Spammers Campaigns in Twitter. *Procedia Comput. Sci.* **2019**, *163*, 248–256. [CrossRef]
24. Antonakaki, D.; Polakis, I.; Athanasopoulos, E.; Ioannidis, S.; Fragopoulou, P. Exploiting abused trending topics to identify spam campaigns in Twitter. *Soc. Netw. Anal. Min.* **2016**, *6*, 48. [CrossRef]
25. Abu-Salih, B.; Qudah, D.A.; Al-Hassan, M.; Ghafari, S.M.; Issa, T.; Aljarah, I.; Beheshti, A.; Alqahtan, S. An Intelligent System for Multi-topic Social Spam Detection in Microblogging. *arXiv* **2022**, arXiv:2201.05203.
26. Koggalahewa, D.; Xu, Y.; Foo, E. An unsupervised method for social network spammer detection based on user information interests. *J. Big Data* **2022**, *9*, 7. [CrossRef]
27. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
28. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT press: Cambridge, MA, USA, 2016; Volume 1.
29. Alharbi, A.R.; Aljaedi, A. Predicting Rogue Content and Arabic Spammers on Twitter. *Future Internet* **2019**, *11*, 229. [CrossRef]
30. Wu, T.; Wen, S.; Liu, S.; Zhang, J.; Xiang, Y.; Alrubaian, M.; Hassan, M.M. Detecting spamming activities in twitter based on deep-learning technique. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e4209. [CrossRef]
31. Ameen, A.K.; Kaya, B. Spam detection in online social networks by deep learning. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018.
32. 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]
33. Mardi, V.; Kini, A.; Sukanya, V.M.; Rachana, S. Text-Based Spam Tweets Detection Using Neural Networks. In *Advances in Computing and Intelligent Systems*; Springer: Singapore, 2020; pp. 401–408.
34. Jain, G.; Sharma, M.; Agarwal, B. Spam detection on social media using semantic convolutional neural network. *Int. J. Knowl. Discov. Bioinform. (IJKDB)* **2018**, *8*, 12–26. [CrossRef]
35. Wu, Y.; Lian, D.; Xu, Y.; Wu, L.; Chen, E. Graph convolutional networks with markov random field reasoning for social spammer detection. *AAAI Conf. Artif. Intell.* **2020**, *34*, 1054–1061. [CrossRef]
36. Li, A.; Qin, Z.; Liu, R.; Yang, Y.; Li, D. Spam review detection with graph convolutional networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing China, 3–7 November 2019.
37. Jain, G.; Sharma, M.; Agarwal, B. Spam detection in social media using convolutional and long short term memory neural network. *Ann. Math. Artif. Intell.* **2019**, *85*, 21–44. [CrossRef]
38. Abozinadah, E.A.; Mbaziira, A.V.; Jones, J.H.J. Detection of abusive accounts with Arabic tweets. *Int. J. Knowl. Eng.-IACSIT* **2015**, *1*, 113–119. [CrossRef]
39. El-Mawass, N.; Alaboodi, S. Detecting Arabic spammers and content polluters on Twitter. In Proceedings of the 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC), Beirut, Lebanon, 21–23 April 2016.
40. Al-Azani, S.; El-Alfy, E.-S.M. Detection of arabic spam tweets using word embedding and machine learning. In Proceedings of the 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, 18–20 November 2018.
41. Boreggah, B.; Alrazooq, A.; Al-Razgan, M.; AlShabib, H. Analysis of Arabic Bot Behaviors. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018.
42. Alshehri, A.; Alhuzali, E.B.N.H.; Abdul-Mageed, M. Think before your click: Data and models for adult content in arabic twitter. In Proceedings of the TA-COS 2018: 2nd Workshop on Text Analytics for Cybersecurity and Online Safety, Miyazaki, Japan, 7–12 May 2018.
43. Mubarak, H.; Abdelali, A.; Hassan, S.; Darwish, K. Spam detection on Arabic twitter. In Proceedings of the International Conference on Social Informatics, Pisa, Italy, 6–9 October 2020; Springer: Cham, Switzerland, 2020.
44. Alom, Z.; Carminati, B.; Ferrari, E. A deep learning model for Twitter spam detection. *Online Soc. Netw. Media* **2020**, *18*, 100079. [CrossRef]
45. Salama, R.A.; Youssef, A.; Fahmy, A. Morphological word embedding for arabic. *Procedia Comput. Sci.* **2018**, *142*, 83–93. [CrossRef]

46. Alsaleh, D.; Larabi-Marie-Sainte, S. Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms. *IEEE Access* **2021**, *9*, 91670–91685. [CrossRef]
47. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Processing Manag.* **2020**, *57*, 102121. [CrossRef]
48. Hegazi, M.O.; Al-Dossari, Y.; Al-Yahy, A.; Al-Sumari, A.; Hilal, A. Preprocessing Arabic text on social media. *Heliyon* **2021**, *7*, e06191. [CrossRef]
49. Twitter, Twitter Help Center. Twitter Rules. 2021. Available online: https://help.twitter.com/en/rules-and-policies/twitter-rules (accessed on 22 June 2021).
50. 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), Perth, Australia, 1–2 September 2011.
51. Wu, C.; Zhang, F.; Xia, J.; Xu, Y.; Li, G.; Xie, J.; Du, Z.; Liu, R. Building Damage Detection Using U-Net with Attention Mechanism from Pre-and Post-Disaster Remote Sensing Datasets. *Remote Sens.* **2021**, *13*, 905. [CrossRef]
52. Abkenar, S.B.; Mahdipour, E.; Jameii, S.M.; Kashani, M.H. A hybrid classification method for Twitter spam detection based on differential evolution and random forest. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6381.
53. Alruily, M. Classification of Arabic Tweets: A Review. *Electronics* **2021**, *10*, 1143. [CrossRef]