MSA UNIVERSITY ENGINEERING JOURNAL

Arabic Tweets Spam Detection Based on Various Supervised Machine Learning and Deep Learning Classifiers

Shimaa I. Hassan[1,a], Mina S. Andraws [1,2,c *], Lamiaa Elrefaei[1,b]

[1] Electrical Engineering Department, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt
[2] Engineering Department, Nuclear Research Centre, Egyptian Atomic Energy Authority, Cairo, Egypt

E-mail: [a]SHAIMAA.RIZK@feng.bu.edu.eg,
[b]LAMIA.ALREFAAI@feng.bu.edu.eg,
[c,*]m.andros56801@feng.bu.edu.eg (Corresponding author)

Abstract

In this paper, different machine learning algorithms, ensemble algorithms, and deep learning algorithms are applied to Arabic tweets to detect whether it human-generated or not. The tweets are used twice as preprocessed and non-preprocessed to measure the effectiveness of Arabic preprocessing in the classification process. The data is also tokenized with various methods like unigram, trigram, and Term Frequency–Inverse Document Frequency. The experiments show that the support vector machine with the non-preprocessed tweets and unigram tokenization has the best performance of 83.11% and a precision of 0.9516 while it predicts the spam or not in a relatively small time.

Keywords: Machine Learning, Ensemble, Deep Learning, Arabic Tweets, Twitter spam.

ENGINEERING JOURNAL Volume 2 Issue 2
Received Date January 2023
Accepted Date March 2023
Published Date March 2023
DOI: 10.21608/MSAENG.2023.291931

1. Introduction

Nowadays reviews on websites, different applications, and social media are a great deal. These reviews reflect the evaluation of the services, the products, and the places. People use these evaluations to decide whether to use this service, buy this product, and reserve this place. These reviews also affect companies, as they design their product features, services, and marketing campaigns according to these reviews. So, opinion-mining tools are designed to assist businesses. These tools are sentiment classification, Featured base opinion-mining comparative sentences, and opinion searches[1]. Although these reviews greatly help customers and companies, it's a double-edged sword. Companies can make limitations on who can review on their applications by pairing them with serial numbers or reservation numbers, but there are no limitations on who can write on social media platforms like Twitter, Facebook, etc. Lately, competition exploits this situation and uses paid attacks that affect business development and people's decisions [2]. These paid attacks are usually made by bots. Are they real reviews or not (spam)?

The detection task has two aspects spam detection and spammer detection. The main task of spam detection is to classify the subjugated text as human-generated or bot-generated [3]. The task of spammer detection is to find the source of spam and whether the spammer is a single spammer or a group spammer. Three techniques can be adopted to detect the spam text or the spammer. The first two can be applied to the text. Natural language processing (NLP) and product feature detection are two techniques. The last one is reviewer behaviour analysis, which detects the reviewer's IP, the review's repeatability, and the review's time [4].

For several reasons, spam in social media can be an issue. It may clog consumers' feeds and make it challenging to locate pertinent and helpful content. Additionally, spam links and comments may contain harmful information that can be exploited to distribute malware or phishing scams. It also may contain hate speech that might increase racial problems in society [5].

This paper contributes to the identification of tweets spam in the Arabic language. The effect of text preprocessing on the identification process has been taken into consideration. The same entities are applied on the same models in both training and testing process twice. One of them is preprocessed and the other isn't. Various classifiers have been applied to identify tweets spam. These classifiers can be grouped into three categories Machine Learning (ML), Ensemble Machine Learning (EML), and Deep Learning (DL). Different NLP feature extraction techniques are used.

The rest of the paper can be described as follows. Section II presents the researchers' previous work to detect spam text. Section III provides some background details on used ML and DL algorithms. Section IV discusses and introduces the result of each used algorithm. Finally, the work conclusion is demonstrated in section V.

2. Related Work

Distinguishing a real review from a bot review becomes essential in modern systems. So, several spam detection techniques have been developed by researchers to discover the authentication of the reviews. That creates the need to extract new features. Review-centric features analyse textual content such as sentiment features, text length, n-grams, skip-grams, linguistic characteristics, word frequency, and bag-of-words. Reviewer-centric features define users' information. These features like interactions, actions, timestamps, text duplication, and text counting. Product-centric features measure the association between the text of the review and the product information [6]. Unfortunately, most of the researchers train their models on certain types of datasets, social media, hotel reviews, economics, and politics. That makes the model centre oriented about a certain topic and doesn't perform well with other topics so using a generic data set is advised. One of the solutions proposed is to create a model that classifies the reviews into different types of fake reviews instead of classification only as fake or real.

The social media reviews might be in Arabic language or English. Arabic spam detector for opinion reviews was introduced [7] to solve this problem. It is designed with techniques from both text mining and data mining. Approaches like support vector machine (SVM) Naive Bayes (NB) and the k nearest neighbour (kNN) are used in classifying the opinion reviews. A new dataset has been created from TripAdvisor, booking, and Agoda. This data wasn't labelled so, the label was assigned through different rules. The data is subjugated to two main preprocessing approaches data preprocessing and text preprocessing. The data preprocessing consists of a few steps. In the first step, the irrelevant attributes have been removed. In the second step, the absent values are changed with the mean of each attribute because of the dataset's small size. In this third step, the continuous attributes are changed to category attributes like age. The distribution of the classes is imbalanced as the data set has only 13.3% of spam records. This issue is overcome through oversampling by duplicating the spam instances to make a balanced data set. The text preprocessing consists of a few steps. Removing the non-Arabic text, tokenization, removing frequent non-content pairing tokens. The NB classifier shows its superiority over the other two classifiers.

The Twitter classifier is designed to detect automated Arabic tweets (bots) [8]. The dataset consists of id, tweet text, and label. It is noticed that the automated tweets tend to be formal Arabic, on the other hand, manual tweets tend to be dialectal Arabic. Four feature groups have been extracted to assess the tweet text: formality, structural, tweet-specific, and temporal features. The formality features consist of three attributes to measure if the tweet contains emotions, diacritics, and four consecutive characters like هههه. The structural features are the count of characters, question marks, and exclamation marks. Tweet-specific features have been proven to be effective in the classification[9]. Its main measures are retweets, hashtags, URLs, and the source field. Temporal features measure the posting nature of the tweets. The text is preprocessed to create a unigram feature vector. The features extracted and the unigram feature vector are used to design the classifier. SVM, NB, and decision tree (DT) classifiers are used to detect the automated tweet. The usage of all the mentioned features shows a great classification performance, especially with SVM and DT.

This researcher defines spam reviews as repetitive, nonrelated, automated, advertising, inappropriate, or malicious URL review [10]. The dataset is collected from posts and comments on Facebook that are interested in social, politics, sports, and music. The data collection focuses on the expression of opinion posts and comments to be used in sentiment analysis. The under-sampling approach is used to solve the problem of unbalanced data. The data set is preprocessed with common NLP techniques like normalization, stemming, tokenization, and stop words removal. The feature extraction process creates nine new features like the number of hashtags, critics, emoticons, lines, etc. Seven models have been introduced to classify the data NB DT, J48, logistic regression, SMO, and LWL.

The ensemble approach is used in Arabic spam detection [11]. Two data sets or used to train these models. The first dataset is the English to Arabic translated dataset. Arabic TripAdvisor, booking and Agoda dataset is The second dataset [7]. To increase the accuracy of the ensemble models few actions have been taken. The first step is to pre-process the data, techniques such as tokenization: transforming the text into a sequence of tokens, non-Arabic text removal, normalization: making each letter appear in its base form, stop words removal, and light stemming where stemming isn't very useful in the Arabic language. The second step is to extract new features. N-gram feature extraction (the unigram, bigram, and trigram) are used. Negation handling reverses the polarity of the world if the previous word is a negation word. Content-based feature extraction like word counts, unique word percentage, and review rating deviation are used. Finally, the last step is to create the detection model. Four models have been designed to detect if the review is spam or not. these models are rule-based classifiers, classic ML classifiers (DT, NB, Logistic Regression, SVM, K-Means, K-NN, Bagging, Boosting, Random Forest (RF), and Neural Networks (NN).), majority voting ensemble classifiers (between rule-based and ML), and stacking ensemble classifiers. A stacking ensemble classifier that combines K-means classifier with a rule-based classifier showed its superiority over other classifiers.

The introduction of DL provides the systems with tools that can learn more features and extract more higher-level features from low-level features. Approaches such as Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and the Transformer.

One of the important research points is the detection of spam text and attached image in the tweet [12]. The dataset is collected automatically from Twitter ABI from Iraqi Arabic tweets. The feature extraction is divided into two phases one for the text and the other for the image. Repeated words, keywords, time of the tweet, length of the tweet, and the WOER2VEC technique are used as text features. The HOGE technique is used for image feature extraction. Convolution neural network (CNN) and SVM algorithms are crossbred for tweets classification as spam or not. The accuracy of the combined algorithm is 98%.

Another DL algorithm is introduced Deep Convolutional Forest (DCF) to detect SMS spam [13]. Each message in the dataset is preprocessed by stemming and stop word removal. Then word vectors are calculated by word embeddings using the GloVe algorithm that vector estimates the word meaning in the message. A Word matrix is constructed for each message from word vectors. The DCF input layer takes the word matrix to generate a feature map. The DCF levels are cascaded where each level gets the processed features from the previous level and pass its output to the next level. The number of levels is fixed but it is a function of the

accuracy of validation data. If the accuracy doesn't meet the predetermined threshold a new level is generated until satisfying the desired accuracy or degrading the performance. Each level consists of a convolution layer, a pooling layer, and the classification layer. The convolution layer convolutes the word matrix to extract features with rectified linear unit (ReLU) activation function. The pooling layer prevents overfitting and complexity by reducing the features passed to the classification layer. The classification layer has two RFs and two extremely randomized trees then it averages the outputs. Different levels of DCF have different types of RFs to provide diversity. The DCF calculates the levels average probabilities of human-generated and spam independently and makes the decision. The DCF has an accuracy of 98.38%.

The transformer is a DL algorithm that works by transforming the text into embeddings and trying to find the relation between embeddings with the attention mechanism [14]. A comparison between different types of ML and DL including transformer will show the huge difference in performance [15]. Two English data sets are used SMS spam dataset and the competition dataset of Kaggle Twitter spam detection. The data is separated into training, validation, and testing 50%, 20%, and 30% respectively. The entries are tokenized after that word embeddings are calculated. For ML algorithms (such as SVM, Logistic regression, NB, and RF), Term Frequency–Inverse Document Frequency (TF-IDF) is used to calculate word embeddings. For DL algorithms (such as LSTM, CNN-LSTM, and transformer), GloVe is used to get vector representations. The Transformer used is a full transformer (encoder + decoder) followed by an activation layer to classify the data. The transformer shows an accuracy of 98% for the first data set, and 87% for the second dataset with a 4% difference from the NB model.

The transformer model creates new capabilities and challenges. The transformer has the ability to understand the context of written text. Although the huge increment in capabilities, the transformer can generate understandable text that can make fake reviews that cannot be detected by traditional means. one of these models is the GPT2 model which creates tweets that are mixed with human-generated tweets to make the dataset. An AraBERT transformer is the same as a BERT transformer but trained specially to deal with the Arabic language. The AraBERT is used to classify these tweets [16]. The human tweets are extracted from the Arabic Twitter dataset [8] and then adding to them tweets from users' timelines to increase the human-generated dataset. This dataset is preprocessed by normalization the text of the tweets, removing non-Arabic characters and punctuation marks, splitting hashtags, URLs, and removing diacritics. Human-generated tweets seed the AraGPT 2 [17] small Arabic that will generate deep fake tweets. The entries of this dataset will be the input to the encoder of pretrained to AraBERT [18] transformer that is followed by a feed-forward classifier layer. That will be trained by 80% of the data. This model shows outstanding performance over other DL models such as LSTM, GRU, biLSTM, and biGru.

Table 1 presents a comparison between different researches. It shows the task solved by each research, the preprocessing, and feature extraction used on the selected datasets. it also presents the algorithms applied to solve the specified task. Finally, the algorithm with the best performance measures is mentioned.

Table 1. comparison of different researches of spam detection

Paper	Task	Preprocessing	Feature extraction	Classification Algorithms	Dataset	Best Results
A. El-Halees et al. 2015 [7]	Detecting Spam in hotels' Arabic reviews.	Removing the non-Arabic text, and removing frequent non-content pairing tokens.	tokenization	SVM, NB, and K-NN	Hotel Arabic Reviews Dataset (HARD) from Booking, Agoda, and TripAdvisor.	NB Accuracy: 99.20%
T. Elsayed et al. 2015 [8]	Detecting spam in Arabic Tweets	Removing the non-Arabic text	formality features emotions, diacritics, and four consecutive characters. structural features: no. of characters, question marks, and exclamation. Tweet-specific features: retweets, hashtags, URLs, and the source field Temporal features: posting nature.	SVM NB DT	Tweets of different Arabic dialects.	Decision Tree J48 with unigram and tweet-specific features Accuracy: 92%
M. Mataoui et al. 2017 [10]	Detecting spam in Facebook posts and comments	Stemming, normalization, and stop-words removal.	A number of lines, hashtags, emoticons, diacritics, specific sequences existence, user publication frequency, repetition frequency of a comment, and similarity between post and comment topics.	NB, SVM, SMO, SGD, DT J48, Decision table, Logistic, and Regression Classifier.	Facebook Posts and comments in the Algerian dialect.	DT J48 Accuracy: .9173 Precision: .926 F_Measure: .955

Paper	Task	Preprocessing	Feature extraction	Classification Algorithms	Dataset	Best Results
R. M. K. Saeed et al. 2019 [11]	Arabic opinion reviews Spam Detection.	Stemming, normalization, and stop-words removal.	N-gram (unigram, unigram and bigram, unigram and bigram and trigram) Negation handling (Polarity of N-gram) Content-based feature extraction: Words count, Unique words percentage, and Review rating deviation.	1.Rule-based classifier 2.ML classifier: NB, SVM, SMO, SGD, DT J48, Logistic Regression, K-Means, KNN, Bagging, Boosting, RF 3. Majority voting ensemble classifier 4. Stacking ensemble classifier: The rule-based then ML classifier.	1. DOSC English to Arabic translated reviews. 2.HARD Arabic dataset [7]	The rule-based with K-Means stacking classifier unigram, bigram, trigram DOSC/HARD Accuracy: 95.25% Recall: 0.9175 Precision:0.9866 F_Measure: 0.9508 / Accuracy 99.98% Recall: 0.9998 Precision:0.9998
O. A. Ismael et al. 2022 [12]	detects Twitter spam text and analyzes images	NS	Text: Repeated words, keywords, time of the tweet, length of the tweet, and WOER2VEC. Image: HOGE	Text: SVM, NB, KNN, DT and RF Image: CNN	Iraqi Arabic tweets	CNN With SVM Accuracy: 98%

Paper	Task	Preprocessing	Feature extraction	Classification Algorithms	Dataset	Best Results
M.A. Shaaban et al. 2022 [13]	Detect Spam in text	stemming and stop words	Word embeddings (GloVe)	Deep Convolutional Forest (DCF) with 2 RF and 2 extremely randomized decision trees	SMS spam collection dataset	DCF with diverse forest (Same forest/ not Same forest) Accuracy: 98.38% Recall: .9111/0.997 Precision:.9880 /0.983 F_Measure:.948 /0.99
X.Liu et al 2021[15]	SMS spam detection	Tokenization	TF-IDF to calculate word embeddings for ML algorithms GloVe to calculate vector representation for DL algorithms (Word embeddings)	Machine learning: Logistic regression, NB, RF and SVM Deep Learning: LSTM, CNN-LSTM, and transformer	1.SMS spam collection dataset 2. Kaggle Twitter spam detection competition dataset.	Transformer 1. Accuracy: 98.92% Recall: .9451 Precision: 0.9781 F_Measure: 0.9613 Accuracy: 87.06% Recall: .8576 Precision: 0.8746 F_Measure: 0.8660
F. Harrag et al. 2021 [16]	Detecting spam in Arabic Tweets	Normalize by removing URLs, splitting hashtags, non-Arabic characters, and diacritics	Word embeddings	The Human tweets is fed to AraGPT2 to generate fake tweets then AraBERT encoder followed by feed-forward classifier layer. LSTM, GRU, biLSTM, and biGru	human tweets from Arabic twitter dataset [8] and users' timeline tweets.	AraBERT then feed-forward classifier layer. Accuracy: 98.7 % Recall: .985 Precision: .989 F_Measure: .987

3. Methodology

In this research, we will apply multiple ML techniques to detect automatically generated Arabic tweets [8]. The dataset consists of 3504 tweets, these tweets are 1560 humans generated tweets and 1944 automated tweets.75% (2628) tweets are used in training and 25% (876) tweets are used in testing. The training and testing entities will be the same for all models. The first step is changing the automated tweets to one and the manual tweets to zero. ML techniques will be applied to this data set without preprocessing as a control. Then ML algorithm will be applied again to this data set after it is preprocessed. The preprocessing will include removing URLs, stop words, English numbers, emojis, standardizing Hamza into a certain form of Hamza, and normalizing lam alif.

Two feature extraction techniques are used. The first one is Tokenization which is applied to the non-preprocessed that set and the preprocessed data set. The second one is Term-Frequency Inverse-Dense-Frequency TF-IDF [19] which is applied to the preprocessed dataset. Tokenization is used by an N-gram technique, where N represents the consecutive words that are taken as one unit. These N-grams are calculated by how many times repeated in the text. Unigram (one word) and Trigram (three words) are used.

The TF-IDF is the multiplication of the two terms. TF represents the count of word repetitions in the text divided by the count of words in the text. IDF is the log of the total text number divided by the count of texts that include the word.

$$TFIDF = Tf\ X\ IDF \tag{1}$$

$$\text{TF} = \frac{count\ of\ word\ repetitions\ in\ the\ text}{count\ of\ words\ in\ the\ text} \tag{2}$$

$$IDF = log\frac{total\ text\ number}{count\ of\ texts\ include\ the\ word} \tag{3}$$

The ML techniques that will be used will be NB, SVM, and DT. The ensemble techniques used are Bagging, RF, and Boosting. The ensemble is classifier combination technique. It performs the classification by calculating the votes from group of base classifiers. These classifiers can be the same or different types. The ensemble classifier performs better when the base classifiers are independent of each other.

3.1. Machine Learning Algorithms (ML)

3.1.1. Naïve Bayes (NB)

It is a simple conditional probabilistic classifier that assumes that all the features are independent of each other and affect the output equally. It determines the likelihood that an event will occur conditioning that a different event had occurred. There are many types of NB like Gaussian, Multinominal, and Bernoulli. The Multinominal Naïve Bayes effectively deals with spam detection problem as the words are independent [20].

$$P(A|B) = \frac{P(A)P(A|B)}{P(B)} \quad (4)$$

3.1.2. Support vector machine (SVM)

It is a ML algorithm that can be used as a supervised classifier or regressor [21]. Its main objective is to classify the data, where its hyperplane has the largest possible margin between classified objects. So mainly SVM has two measures misclassification of points and the margin width.

The equation of the separating line or hyperplane

$$\vec{w} \bullet \vec{x} + b = 0 \quad (5)$$

Where the margin of separation can be represented by the following equations

$$y_i = \begin{cases} 1 & if\ \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & if\ \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases} \quad (6)$$

$$Margin\ = \frac{2}{||\vec{w}||} \quad (7)$$

$$\vec{w} = \sum_i^N \alpha_i\, x \quad (8)$$

So, to obtain the maximum possible margin to separation process we need to minimize w.

$$L(\vec{w}) = \frac{||\vec{w}||^2}{2} \quad (9)$$

If the problem isn't linearly separable, radial basis function (RBF) kernel can be used. It is based on the idea of finding the hyperplane in a high-dimensional space that maximally separates the different classes. this term will be added with $k \geq 2$. Where γ is a hyperparameter that controls the width of the kernel and C is a hyperparameter that controls the trade-off between maximizing the margin and minimizing the classification error, and n is the number of training samples.

$$\mathrm{L}(\vec{\mathrm{w}}) = \frac{||\vec{\mathrm{w}}||^2}{2} + \mathrm{C}\left(\sum_{\mathrm{i}=1}^{\mathrm{N}} \gamma_{\mathrm{i}}^{\mathrm{k}}\right) \quad (10)$$

3.1.3. Decision Tree (DT)

It is a simple classification and prediction algorithm yet powerful. its main idea is to choose a certain feature and separate the entities according to the features outcome. This process is repeated with different features until splitting all the data of the same class or it early terminated by satisfying the classification requirement [21]. The first chosen feature is

called the root. Each feature selected afterward is called a node. The feature reduces the impurity more (with the highest gain) is selected first. The impurity is measured by Gini equation [22]. where P: is presplitting impurity, M: is post splitting impurity, $\boldsymbol{p_i(t)}$: is the frequency of class i at node t, and C:is the total number of classes.

$$Gain = P - M. \tag{11}$$

$$Gini = 1 - \sum_{i=0}^{c-1} p_i(t)^2 \tag{12}$$

3.2. Ensemble algorithm

3.2.1.Bagging

It is called bootstrap aggregation. The same classifier is used as base classifiers but each one is trained with different instances (bags). These instances are chosen randomly with replacement. The instances ratio equation n% for each classifier from n instances.

$$n\% = 1 - (1 - 1/n)n \tag{13}$$

The vote of the classifiers f^* is calculated where $f_i(x)$: is base classifier decision, δ: is1 for true argument and 0 otherwise.

$$f^*(x) = argmax \sum_i \delta(f_i(x) = y) \tag{14}$$

The classifier can be any type, but decision trees commonly used [23]. When they are used each tree is only stump. The bagging model used has 100 stumps as base model.

3.2.2.Random Forest (RF)

It is one of ensemble methods, that have similarity to Bagging. it calculates the prediction from different DT each of them has its own criteria. Also, each DT can use different instances and attributes for each one these trees. These decision trees are unpruned that means they are trained until all nodes are pure. Then the RF algorithm takes the average of outputs in case of regression or the majority vote in case of classification and considered that as the output [23]. The RF model used has 100 unburned trees. Each of them takes 60% random samples the training set.

3.2.3.Boosting

It is one of ensemble methods, that have similarity to Bagging. it calculates the prediction from different DT each of them has its own criteria. Also, each DT can use different instances and attributes for each one these trees. These decision trees are unpruned that means they are trained until all nodes are pure. Then the RF algorithm takes the average of outputs in case of regression or the majority vote in case of classification and considered that as the

output [23]. The RF model used has 100 unburned trees. Each of them takes 60% random samples the training set.

3.3. Deep Learning Algorithm (DL)

3.3.1.Long Short-Term Memory (LSTM)

LSTM is a type of RNN that is often used in NLP and time series forecasting [24]. Unlike traditional RNNs, which can have difficulty learning long-term dependencies in the data, LSTM networks are able to learn and make predictions based on data with long-term temporal dependencies.

LSTM networks are designed to remember information for long periods of time, which is why they are called long short-term memory networks. They have input layer, forget layer, memory cell, and output layer. The input layer receives the input sequence. The forget layer, decides which information from the previous state to discard. The memory cell stores the information that is passed from one time step to the next, so it stores and retrieve information over long periods of time. The output layer, which produces the output at the current time step [25].

In an LSTM network, the memory cells are organized into a series of layers that form a memory cell state that can be updated and passed from one layer to the next. This allows the network to maintain a memory of the data that it has seen and use that memory to make predictions [26].

The used LSTM model consists of five layers. Input layer fed with word tokens followed by the LSTM layer. The LSTM layer has 50 units. The next three layers are dense layers 64, other dense layer 32 and the output layer.

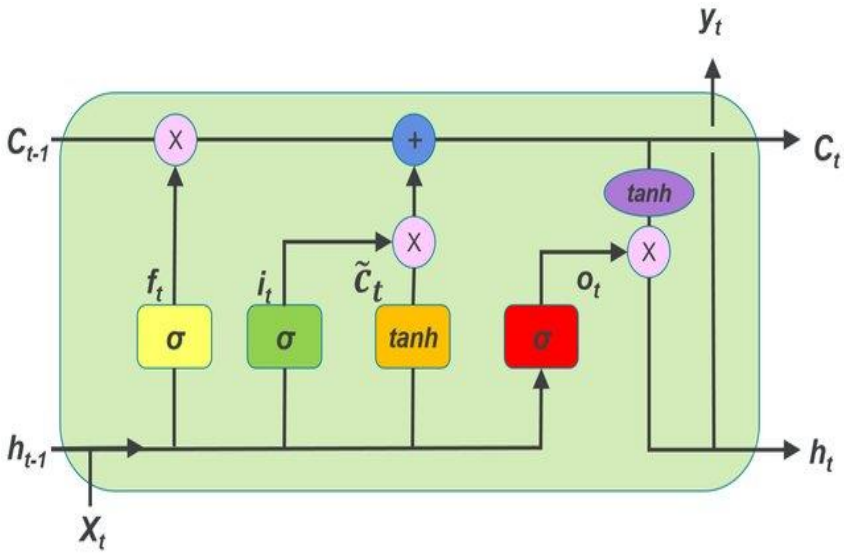

Fig. 1. LSTM unit [25].

3.3.2. Convolution Neural Network (CNN)

CNN are a type of DL algorithm. Although they are more commonly used in computer vision and image recognition, they can also be used in NLP. It consists of few layers convolution layer, max pooling layer, and fully connected layer. Convolutional layers apply a set of filters to the input data and extract features from it. The filters slide over the input data, computing dot products between their weights and the input data at each position. These dot products are then transformed by a ReLU function, before being passed to the next layer. Pooling layers down-sample the data by taking the maximum or average value within a fixed-size window. This helps to reduce the size of the data and reduce the number of parameters in the model, which can improve its generalization ability. Fully connected layer performs classification or regression on the data. The output of the fully connected layers is a set of predictions or class probabilities, depending on the task [27].

In NLP, a CNN can be used to process text data, such as sentences or documents, in a similar way to how they are used to process images. The input data is typically a sequence of words, which is converted into a numerical representation, such as a sequence of word vectors or a matrix of word embeddings. The convolutional layers in the CNN then apply filters to this representation, which are used to detect different patterns or features in the data. These filters are learned by the network through training and can be used to detect patterns such as n-grams, or sequences of words that occur frequently in the data [26].

The output of the CNN is a set of feature maps, which represent the input text in a more abstract and compact form. These feature maps are then fed into a fully connected layer, which uses the feature maps to make predictions about the text. For example, in sentiment analysis, the fully connected layer might predict the sentiment of a sentence or document, based on the patterns of words and features that were detected by the convolutional layers.

Two CNN models are constructed to detect spam. the first model is composed of input layer, convolution layer, max pooling layer, fully connected layer and finally the out layer. The second model consists of seven layers, the first model is composed of input layer, convolution layer, max pooling layer. Then it followed by convolution layer, max pooling layer, fully connected layer the out layer.

3.3.3. Convolution Neural Network Long Short-Term Memory (CNN-LSTM)

CNN-LSTM are a type of DL algorithm that combines the strengths of CNNs and RNNs. They are often used in NLP and computer vision, where they can learn to detect patterns in both spatial and temporal data [28].

CNN-LSTM are called long-term because they are designed to learn and make predictions based on data with long-term dependencies. This is achieved by combining the convolutional layers of a CNN, which are used to detect spatial patterns in the data, with the recurrent layers of an RNN, which are used to learn and remember long-term dependencies in the data.

In an CNN-LSTM, the input data is typically a sequence of tokens, which is processed by the convolutional layers of the network to extract spatial features from the text. These spatial features are then passed through the recurrent layers of the network, which learn to detect temporal dependencies in the data and make predictions based on the long-term context of the input [29].

Two CNN-LSTM models are used. The first model consists of seven layers, they are input layer, and CNN layers (convolution layer, max pooling layer). Then LSTM layer follows them. Four dense layers, and the output layer ae used at the end. The Second model consists of ten layers they are input layer, and two CNN layers (convolution layer, max pooling layer). Then LSTM layer follows them. Three dense layers. and the output layer ae used at the end.

4. Results and Discussion

In this Experiment, the Arabic twitter dataset is divided into training and testing. This dataset is applied two times for ML algorithms where one of them with preprocessing, and the other one without. For DL algorithms, preprocessed dataset is used All the algorithms performance mesures can be seen at Table 2. The non-preprocessed (np) data is tokenized once using unigram, and the other time unigram and trigram together. The performance of Linear SVM with unigram tokenization is superior with 83.11% and 0.9516 precision to other algorithms that trained on non-preprocessed data. The SVM with rpf Kernel has the same results as linear SVM.

The preprocessed data is subjugated to different preprocessing steps such as removing URIs, emojis, English numbers, and stop words. Light stemming, alif lam normalization, and hamza normalization are used to pre-process the text. The preprocessed data is used four times with ML algorithms and Ensemble algorithms. The first time the text is tokenized with unigram only. The unigram and trigram are used in second time. both Linear SVM and rbf Kernel SVM tokenized with unigram and trigram have 82.53% accuracy and 0.9534 precision. The bagging model have the same result.

The third time TF-IDF with unigram is applied to the data. Finally, the TF-IDF with both unigram and trigram is applied to the data. The RF with TF-IDF (unigram + trigram) has the best results of algorithms applied on preprocessed dataset. It has an accuracy of 82.76% and precision of 0.9370.

The DL algorithms applied on preprocessed data using unigram word tokenization. The CNN-LSTM model with two convolution layers has best results between the DL algorithms. Its accuracy is 82.65% with precision of 0.8609.

The prediction time of models have more than 724 true predictions out of 876 test samples is shown in Fig. 2 except for non-preprocessed bagging model as prediction time is 49.703. The non-preprocessed linear SVM and the RF with TF-IDF feature extraction has the best prediction time which is important aspect. Fig. 3 shows the accuracy and precision of algorithms with best performance. The precision also is important aspect as it represents the

number of true tweets that misclassified. The precision of linear SVM is 0.9516 and RF is 0.9370 as they can be calculated from confusion matrices Fig. 4 a and b respectively.

Table 2. Algorithms Performance measures

Model	Prediction Time	Train accuracy	Test accuracy	Precision	Recall	F1 measure
Non-preprocessed						
MlNB(Uni_NP)	0.0010	0.9513	0.7660	0.7751	0.8120	0.7931
MNB(Uni+Tri_NP)	0.0034	0.9749	0.7637	0.7743	0.8079	0.7906
SVM(Uni_NP)	0.9101	0.9395	0.8311	0.9516	0.7314	0.8271
SVM(Uni+Tri_NP)	1.3640	0.9467	0.8276	0.9537	0.7231	0.8225
SVM(rbf_Uni_NP)	1.1420	0.9395	0.8311	0.9516	0.7314	0.8271
DT(Uni_NP)	0.0031	0.9821	0.8059	0.8668	0.7665	0.8135
DT(Tri_NP)	0.0036	0.9821	0.8116	0.8735	0.7707	0.818
RF(Uni_NP)	0.3311	0.9821	0.8231	0.9295	0.7355	0.8212
RF(Uni+Tri_NP)	0.6721	0.9821	0.8162	0.9499	0.7045	0.8090
AdaBoost(Uni_NP)	0.1298	0.8763	0.8242	0.8966	0.7707	0.8288
AdaBoost(Uni+Tri_NP)	0.2086	0.8782	0.8208	0.8902	0.7707	0.8261
Bag(Uni_NP)	49.7037	0.9452	0.8299	0.9443	0.7355	0.8269
Bag(Uni+Tri_NP)	76.7820	0.9494	0.8253	0.9534	0.7190	0.8197
Preprocessed						
MlNB(Uni)	0.0011	0.9536	0.7660	0.7751	0.8120	0.7931
MNB(Uni+Tri)	0.0025	0.9734	0.7603	0.7708	0.8058	0.7878
SVM(Uni)	0.6803	0.9456	0.8242	0.9412	0.7273	0.8205
SVM(Uni+Tri)	1.0141	0.9494	0.8253	0.9534	0.7190	0.8197
SVM(rbf_Tri)	1.0151	0.9494	0.8253	0.9534	0.7190	0.8197
DT(Uni)	0.0022	0.9821	0.8105	0.8502	0.7975	0.8230
DT(uni+Tri)	0.0025	0.9821	0.8002	0.8519	0.7727	0.8104
MNB(TFIDF+Uni)	0.0020	0.9566	0.7466	0.7399	0.8347	0.7844
MNB(TFIDF+Uni+Tri)	0.0028	0.9722	0.7363	0.7192	0.8574	0.7822
SVM(TFIDF+Uni)	0.6703	0.9775	0.8242	0.9609	0.7107	0.8171
SVM(TFIDF+Uni+Tri)	1.0012	0.9806	0.8208	0.9739	0.6942	0.8106
DT(TFIDF+Uni)	0.0025	0.9821	0.8082	0.8405	0.8058	0.8227
DT(TFIDF+Uni+Tri)	0.0027	0.9821	0.8139	0.8527	0.8017	0.8264
RF(Uni)	0.2631	0.9817	0.8139	0.9002	0.7459	0.8158
RF(Uni+Tri)	0.3465	0.9821	0.8151	0.9328	0.7169	0.8107
AdaBoost(Uni)	0.1140	0.8786	0.8162	0.9007	0.7500	0.8184
AdaBoost(Uni+Tri)	0.1304	0.8774	0.8151	0.8908	0.7583	0.8191
Bag(Uni)	43.2226	0.9482	0.8253	0.9437	0.7273	0.8214
Bag(Uni+Tri)	65.1723	0.9524	0.8253	0.9534	0.7190	0.8197
RF(TFIDF+Uni)	0.2632	0.9821	0.8174	0.9050	0.7479	0.8190
RF(TFIDF+Uni+Tri)	0.3416	0.9817	0.8276	0.9370	0.7376	0.8254
AdaBoost(TFIDF+Uni)	0.1010	0.8888	0.8014	0.8429	0.7872	0.8141
AdaBoost(TFIDF+Tri)	0.1325	0.8896	0.8116	0.8584	0.7893	0.8223
Bag(TFIDF+Uni)	50.5933	0.9775	0.8196	0.9478	0.7128	0.8136
Bag(TFIDF+Uni+Tri)	64.9410	0.9798	0.8208	0.9685	0.6983	0.8115

Model	Prediction Time	Train accuracy	Test accuracy	Precision	Recall	F1 measure
Deep Learning						
LSTM	3.6488	0.9273	0.8208	0.8724	0.7913	0.8299
CNN(1)	3.0708	0.9402	0.8231	0.8647	0.8058	0.8342
CNN (2)	3.0647	0.9542	0.8151	0.8531	0.8037	0.8276
CNN-LSTM (1)	100.2481	0.9456	0.8184	0.8406	0.8285	0.8345
CNN-LSTM (2)	3.8212	0.9475	0.8265	0.8609	0.8182	0.8389

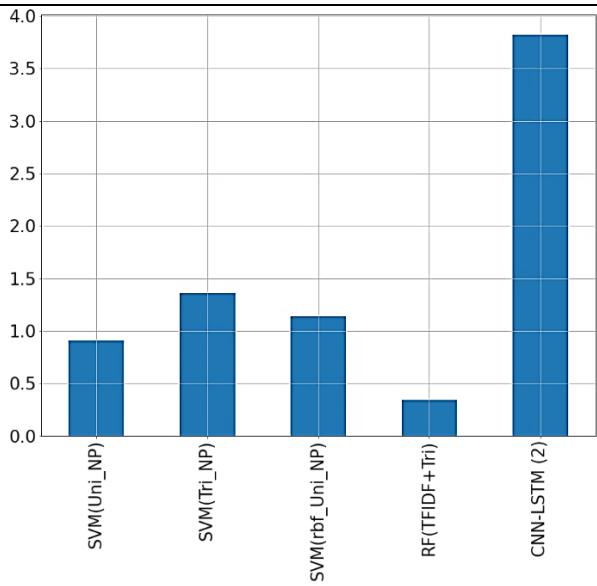

Fig. 2. Prediction time of best performance algorithms

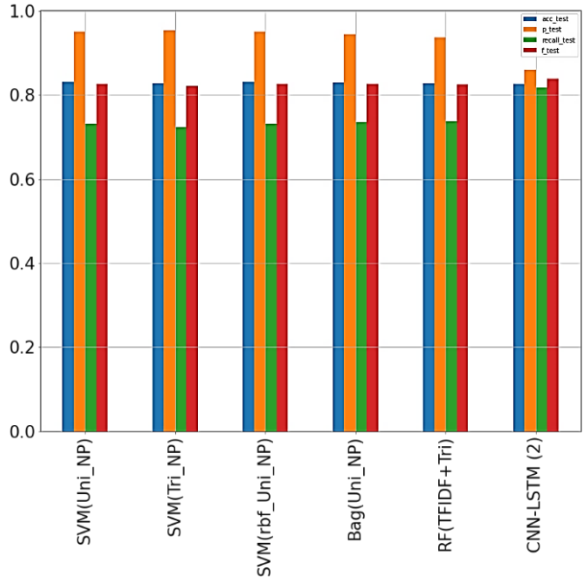

Fig. 3. Best models accuacy and percision

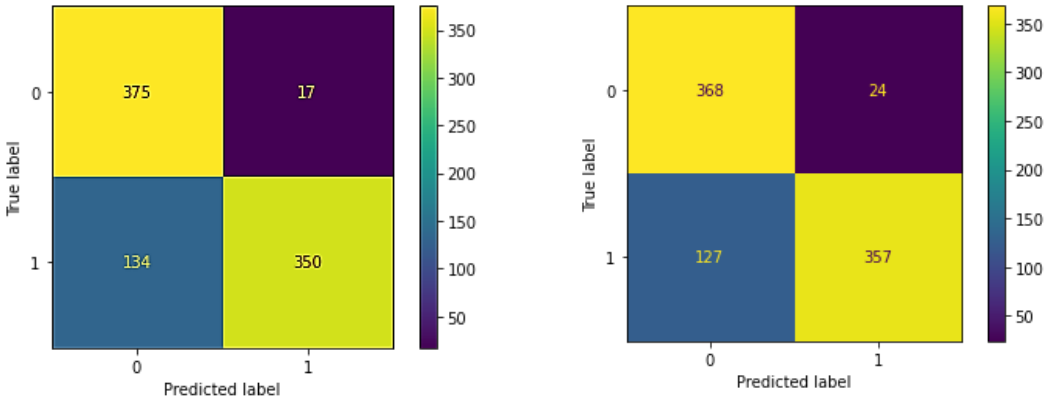

Fig 4. Confusion Matrix a) NP Linear SVM. b) RF (TF-IDF+Tri) Confusion Matrix

The best performed model is compared with other model that used the same dataset the comparison can be shown Table 3. The other model used several feature extraction Techniques formality features (F) such as emotions, diacritics, and four consecutive characters. structural features (S) like the count of characters, question marks, and exclamation. Tweet-specific (TS) features as retweets, hashtags, URLs, and the source field. The temporal features (temp) are the time and the posting nature. then apply them individually and then combined together.

Table 3. Result comparison

Researcher	Notes	Models	Accuracy
T. Elsayed et al. 2015 [8]	non-preprocessed tweets. U	SVM	60%
		DT J48	61.2%
		MNB	61.3%
	Preprocessed (removing non-Arabic text) U+TS	SVM	89.2%
		DT J48	91.97%
		MNB	86.1%
	Preprocessed (removing non-Arabic text) U+F	SVM	62%
		DT J48	69.67%
		MNB	49.1%
	Preprocessed (removing non-Arabic text) U+S	SVM	62%
		DT J48	84.4%
		MNB	66.67%
	Preprocessed (removing non-Arabic text) U+Temp	SVM	66.67%
		DT J48	87.3%
		MNB	65.1%
	Preprocessed (removing non-Arabic text) U+TS +S+F+Temp	SVM	89.2%
		DT J48	86.5%
		MNB	82.8%
Our Model	non-preprocessed tweets. U	SVM	83.11%
		DT	80.59%
		MNB	76.6%

5. Conclusion

The heading o This paper applies different classification algorithms such as ML, ensemble, and DL. The ML algorithms used NB, SVM and DT. The ensemble algorithms that applied are bagging with SVM as base model, boosting and RF with DT as base model. The DL models are LSTM, CNN, and LSTM-CNN. These algorithms are used for detection of Arabic tweets spam. The text of the tweets is used as preprocessed and non-preprocessed with these algorithms. Few algorithms show a performance that is near to each other, but SVM with non-preprocessed text tokenized with unigram shows the best performance. Its accuracy has 83.11% with precision of 0.9516. Th importance of precision is that it will be a great deal to classify a human generated tweet as spam. The SVM classifies the output in relatively good time and that is important in heavy traffic social medial like Twitter.

Acknowledgement

I would like to thank associate prof. Tamer Elsayed for providing the used dataset.

Reference

[1] A. Heydari, M. a. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634-3642, 2015/05/01/ 2015, doi: https://doi.org/10.1016/j.eswa.2014.12.029.

[2] E. Ferrara, "Measuring social spam and the effect of bots on information diffusion in social media," in *Complex spreading phenomena in social systems*: Springer, 2018, pp. 229-255.

[3] S. Rao, A. K. Verma, and T. Bhatia, "A review on social spam detection: challenges, open issues, and future directions," *Expert Systems with Applications*, vol. 186, p. 115742, 2021.

[4] M. Alruily, "Classification of Arabic Tweets: A Review," *Electronics*, vol. 10, no. 10, p. 1143, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/10/1143.

[5] R. Bailurkar and N. Raul, "Detecting Bots to Distinguish Hate Speech on Social Media," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2021: IEEE, pp. 1-5.

[6] A. Z. Mohammed Ennaouri, "Fake Reviews Detection through Machine learning Algorithms: A Systematic Literature Review," 2022, doi: https://doi.org/10.21203/rs.3.rs-2039197/v1.

[7] A. El-Halees and A. Hammad, "An Approach for Detecting Spam in Arabic Opinion Reviews," *International Arab Journal of Information Technology*, vol. 12, 01/01 2015.

[8] H. Almerekhi and T. Elsayed, "Detecting Automatically-Generated Arabic Tweets," in *Information Retrieval Technology*, Cham, G. Zuccon, S. Geva, H. Joho, F. Scholer, A. Sun, and P. Zhang, Eds., 2015// 2015: Springer International Publishing, pp. 123-134.

[9] M. Hasanain, T. Elsayed, and W. Magdy, "Identification of Answer-Seeking Questions in Arabic Microblogs," 2014, pp. 1839-1842, doi: 10.1145/2661829.2661959.

[10] M. Mataoui, O. Zelmati, D. Boughaci, M. Chaouche, and F. Lagoug, "A proposed spam detection approach for Arabic social networks content," in *2017 International*

Conference on Mathematics and Information Technology (ICMIT), 4-5 Dec. 2017 2017, pp. 222-226, doi: 10.1109/MATHIT.2017.8259721.
[11] R. M. K. Saeed, S. Rady, and T. F. Gharib, "An ensemble approach for spam detection in Arabic opinion texts," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 1, pp. 1407-1416, 2022/01/01/ 2019, doi: https://doi.org/10.1016/j.jksuci.2019.10.002.
[12] O. A. Ismael and Ö. Ü. Y. Çelik, "A new approach to arabic spam tweet detection in Twitter using machine learning algorithms," *AIP Conference Proceedings*, vol. 2398, no. 1, p. 050014, 2022, doi: 10.1063/5.0094050.
[13] M. A. Shaaban, Y. F. Hassan, and S. K. Guirguis, "Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text," (in eng), *Complex Intell Systems*, vol. 8, no. 6, pp. 4897-4909, 2022, doi: 10.1007/s40747-022-00741-6.
[14] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[15] X. Liu, H. Lu, and A. Nayak, "A Spam Transformer Model for SMS Spam Detection," *IEEE Access*, vol. 9, pp. 80253-80263, 2021, doi: 10.1109/ACCESS.2021.3081479.
[16] F. Harrag, M. Debbah, K. Darwish, and A. Abdelali, "BERT Transformer model for Detecting Arabic GPT2 Auto-Generated Tweets," presented at the ICNLSP 2018: 2nd International Conference on Natural Language and Speech Processing, 2021.
[17] W. Antoun, F. Baly, and H. M. Hajj, "AraGPT2: Pre-Trained Transformer for Arabic Language Generation," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop (WANLP)*, Kyiv, Ukraine, 2021: Association for Computational Linguistics, pp. 196–207.
[18] F. B. Wissam Antoun, and Hazem Hajj., "AraBERT: Transformer-based Model for Arabic Language Understanding," in *European Language Resource Association*, Marseille, France, 2020, pp. 9–15.
[19] M. Kantrowitz, B. Mohit, and V. Mittal, "Stemming and its effects on TFIDF ranking," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000, pp. 357-359.
[20] M. Vangelis, A. Ion, and P. Geogios, "Spam filtering with naive bayes-which naive bayes?," in *Third conference on email and anti-spam (CEAS)*, 2006.
[21] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, "A comparative study of spam SMS detection using machine learning classifiers," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 2018: IEEE, pp. 1-7.
[22] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, Second ed. Pearson Education 2019.
[23] V. Gupta, A. Mehta, A. Goel, U. Dixit, and A. C. Pandey, "Spam detection using ensemble learning," in *Harmony search and nature inspired optimization algorithms*: Springer, 2019, pp. 661-668.
[24] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A Clockwork RNN," in *International Conference on Machine Learning*, 2014: PMLR, pp. 1863-1871.
[25] A. A. Ismail, T. Wood, and H. C. Bravo, "Improving Long-Horizon Forecasts with Expectation-Biased LSTM Networks," *ArXiv*, vol. abs/1804.06776, 2018.
[26] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS spam," *Future Generation Computer Systems*, vol. 102, pp. 524-533, 2020.

[27] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 3, pp. 147-156, 1993.
[28] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A C-LSTM neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.
[29] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, "A hybrid CNN-LSTM model for SMS spam detection in Arabic and english messages," *Future Internet*, vol. 12, no. 9, p. 156, 2020.