Ingénierie des Systèmes d'Information
Vol. 29, No. 1, February, 2024, pp. 117-123
Journal homepage: http://iieta.org/journals/isi

A Hybrid Deep Learning Approach for Spam Detection in Twitter

Hemza Loucif

Laboratory of Informatics and its Applications, University of M'sila, M'sila 28020, Algeria

Corresponding Author Email: hemza.loucif@univ-msila.dz

https://doi.org/10.18280/isi.290113

Received: 3 September 2023
Revised: 30 November 2023
Accepted: 5 December 2023
Available online: 27 February 2024

Keywords:
spam, deep neural networks, CNN, PCA, Twitter, Anomaly detection, text classification, social networks, cybersecurity

ABSTRACT

Detecting malicious user accounts on Twitter has become an active area of research in social network analysis. This kind of ill-intentioned users send undesired tweets to other users to promote products, services, rumors, fake news, or any abusive content. Hence, the detection of those spammers and their originators will prevent deterioration in the quality of communication services and legitimate users from being affected. Traditional machine learning techniques have been proposed to tackle the problem of spammers detection. However, many researchers have pointed out that the majority of machine learning based models that rely on supervised classification didn't perform well in noisy and short message platforms like Twitter. Recently, deep learning-based alternatives have shown remarkable performance in this area because of their competitive training speed and low implementation cost. In this paper, we propose a new hybrid architecture that combines Principal Component Analysis (PCA) with Convolutional Neural Network (CNN) to give birth to a more reliable and robust model for spammers detection in Twitter. Unlike other hybridizations, the convolutional layer in the CNN module is not fed traditionally by raw feature vectors, rather, we use very low dimensional vectors containing high-order features provided by PCA module. A series of nicely conducted experiments over benchmark datasets have shown that the hybridization proved to be effective for the detection of spammers. The results show that PCA-CNN model can achieve better classification performance with 94.91% precision, 96.76% recall, and 95.83% F-score when compared to baseline benchmarks like CNN, ANN and SVM.

1. INTRODUCTION

Nowadays, online social networking sites (OSNs) such as Facebook, Instagram, etc. have become the most favorable broadcast medium for people to spend their time and to obtain real-time information. However, the popularity of those communication platforms attracts malevolent entities (aka spammers) to pollute their social environments, and infiltrate legitimate accounts with unsolicited spam contents. Although the diversity of social networks, spammers can easily change their tactics according to the specific nature and population composition of each social platform.

Twitter has come up as one of the most popular large-scale conversational social platforms that allows people to communicate with one another in real time via short 280-character text messages, called tweets [1]. According to Statista research department [2], Twitter's audience accounted for more than 368 million active users at present. The statistics reveal also that this significant microblogging service is dominating social, political, business, and many more affairs in nearly every corner of the world. However, Twitter have become recently a favorite place to spammers for engaging aggressive, deceptive, or bulk behaviors that mislead others and disrupt their experience on this platform. From recent rigorous analysis [3], it is found that about 19.42% of active Twitter accounts are spammy accounts whether they are phishers, promoters, or fake users [4, 5]. Hence, it is not surprising that thousands of accounts are banned monthly for breaking the Twitter Rules. To help the Twitter's community in combating the spam phenomenon, a lot of research has been done to filter out spamming behavior [6, 7]. In this paper, we contribute in fighting spammers by proposing a new simplistic and robust classification algorithm that allows the identification of malicious from legitimate user accounts in Twitter.

The remainder of this paper is organized as follows. Section 2 presents the related work in Twitter spam detection research area. The proposed model that includes two algorithms: PCA and CNN is presented in section 3. Next, we present the experiment results and we describe how the model is evaluated and compared with two benchmarks in section 4. Finally, we summarize the contribution in section 5.

2. RELATED WORK

In the last decade, there has been an increased focus by the social network analysis community on developing innovative techniques to detect spam contents and malicious accounts in social media networks [8]. Elmendili et al. [9] Suggested that account-based spam detection approaches are not best suited for filtering tweets in a real-time detection. Therefore, they proposed a three-parts hybrid system architecture to detect spam tweets on Twitter which consists of a security layer based on social honeypots, a security layer based on content filtering, and a classification layer. The authors use honeypot tweets as a bait to attract malicious profiles and then to collect their characteristics. Jain et al. [10] Following the same approach used by Tai et al. [11] to model tree-structured topologies for semantic sentence modeling, have proposed a deep learning neural architecture by stacking in a sequential manner both Convolutional Neural Network (CNN) and Long Short-Term Neural Network (LSTM). The main contribution in this work is the addition of a new semantic layer just before the embedding layer that captures semantic information to enhance the word representation. In this layer, the semantic of the input text is included over its word2vec based representation by using the knowledge-bases WordNet and ConceptNet. Improvement in the embedding space coverage with better initialization of word vectors are proved to be important to improve the performance of deep learning architectures in the detection of spammers in short and noisy text platforms like Twitter [12]. To detect spams in Twitter, Gharge and Chavan [13] proposed a new architecture that focuses on the analysis of the tweets instead of the user accounts. The architecture consists of five processes, namely, Tweets collection, spam labelling, feature extraction, classification, and spam detection. Because it has been proven that it gave more accurate results than other existing classifiers, the authors have applied the Support vector machine classifier using a prepared setup of Weka. The results showed that the proposed architecture has accurately classified 95-97% of the dataset tweets. Using deep learning, Shahariar et al. [14] proposed a novel model to detect spam text reviews using both labeled and unlabeled data. This model is a mixture of Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Multi-Layer Perceptron (MLP). This model comprises four phases, namely, (1) data acquisition and preprocessing using both labeled and unlabeled datasets, (2) Active Learning for labeling unlabeled data, (3) feature selection using TF-IDF, n-grams and Word2Vec, (4) spam detection through the application of deep learning techniques such as CNN and LSTM. After comparing those technics with some traditional machine learning classifiers such as Naive Bayes (NB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM), it has been shown that deep learning classifiers perform better. With a hybrid approach, that exploits a mixture of community-based features with interaction-based features, Fazil and Abulaish [15] have tried to detect automated spammers in Twitter. The authors have built their approach using three classical machine learning classifiers, namely, decision tree, Bayesian network, and random forest. The results suggested that community - and interaction -based features are proven to be more effective for the detection of spammers in Twitter.

In this work, our main objective is to propose a new simplistic and accurate machine learning based model that we name PCA-CNN, in which we stack together two classical algorithms, namely, the Convolutional Neural Network (CNN) [16, 17] and Principal Component Analysis (PCA) [18, 19] in the way that we can outstand the performance of traditional classifiers.

3. PROPOSED MODEL

This section reveals how this work contributes to move ahead in the fight against spammers in Twitter. The proposed model will enable automatic classification of Twitter accounts into spammers or nonspammers. In order to reach this purpose, our model is implemented based on a combination of Principal Component Analysis (PCA) and Convolutional Neural Networks (CNNs) algorithms. Our choice of this mixture was not random, but it was carefully thought out. Firstly, CNNs are one of the most efficient machine learning classifiers that provide highly accurate results without severe computational complexities or costs. Recent literature reveals the increasing tendency toward large-scale exploitation of CNNs from a broad range of domain applications. Their simplicity, ease of implementation, and empirical success are the main factors that make them highly suitable for our approach. The CNN's architecture which is inspired from the complex connectivity structure of the human brain enables the learning of multiple features from the input data. The hidden layers of the CNN perform feature extraction over the input data by carrying out different manipulations and calculations. In a standard convolution neural network architecture, we can distinguish four important layers that help in extracting information from the input matrix, namely:

- Convolution Layer: It is the first step in the process of extracting relevant features from the input matrix. The CNN slides several filter matrices with different dimensions over the input matrix to perform dot product in order to shape the convolved feature maps.
- ReLU layer: Once the input matrix is filtered and the feature maps are produced, the next step is to forward them to a ReLU (Rectified Linear Unit) activation function. ReLU for short performs an element-wise operation which outputs the input directly if it is positive and sets all the negative values to 0.
- Pooling Layer: In this step, the dimensionality of the rectified feature map is reduced, which leads to a real compression to the result that has been received from the precedent layer. Max pooling and average pooling are the most popular pooling filters that are usually used for generating pooled feature maps. After taking the pooling step, flattening is the next step through which all the obtained 2-Dimensional pooled feature maps will be converted into 1-Dimensional continuous linear vectors.
- Fully-connected layer: After turning the pooled feature maps into a single long column vector, we pass it down to a fully connected neural network for classification to get the final output.

Secondly, PCA is a workhorse algorithm in statistics, where dominant correlation patterns can be extracted from high-dimensional data. PCA, is also the bedrock dimensionality reduction technique which is still very commonly used in data science and machine learning applications to reduce the dimensions of large data sets. This technique can effectively transform a large number of variables that exist in a dataset into a smaller one that still preserve most of the information in that dataset. PCA can be broken down into five steps as follows:

- Step 1 (Standardization): For making sure that all the features of the dataset are measured on the same scale, we resort to standardize the range of the variables so that they contribute all equally to the analysis and don’t lead to biased results. Standardization (or Z-score normalization) can be done mathematically by subtracting the mean for each value of each variable and then dividing by the standard deviation [20]. Once the standardization is done, all the variables are all brought to the same scale.

- Step 2 (Covariance Matrix computation): In this step, we check whether there exists any dependence between the variables (i.e., features) of the input dataset. Covariance matrix is the tool that is used to identify these dependencies and quantify the existing correlations between the set of measured variables. For k-dimensional data, the covariance matrix is a k×k symmetric matrix where the elements represent the covariances between all possible pairs of features. According to the sign of the covariance between a couple of variables, we say that they are correlated (they increase or decrease together) if the sign is positive, and vice versa.
- Step 3 (Eigenvalues and eigenvectors computation): In this step, we determine a new set of axes (i.e., directions) along which the data varies the most which are the dominant eigenvectors that we need to compute from the covariance matrix. Let C be the covariance matrix (a square symmetric matrix), ν a non-zero vector, and λ a scalar. If those elements satisfy $C\nu=\lambda\nu$, then we can say that λ is the eigenvalue which is associated with the eigenvector ν of the covariance matrix C.
- Step 4 (relevant features selection): Once the eigenvector components have been computed, we will get a list of principal components respecting the descending order of the eigenvalues. After arranging the eigenvectors in the order of their associated eigenvalues, we pick up the first topmost eigenvectors to determine the principal components that represent the new uncorrelated variables (i.e., dimensions) that capture the maximal amount of variance (i.e., information) of our data. Hence, by discarding all the less significant components (with low eigenvalues), we can consequently reduce the dimensionality of our data set and just keep the most relevant features without losing much information.

In comparison to most previous works that used CNN-based approaches, our modal does not feed raw feature vectors directly to the core building block of CNN which is the convolutional layer. Rather, we use very low dimensional vectors containing high-order features which are computed using PCA. The benefit in using PCA is twofold:

(1) Dimensionality reduction: given a high dimensional data, PCA transformation will keep only the relevant fraction of the extracted features. It is a simple and powerful mechanism that reduces noise in our dataset.

(2) Classification accuracy improvement: minimizing the initial number of features from over 200 to just a small fraction of principal components would simplify the task of the CNN component and certainly maximize the classification performance.

In the proposed model as shown in Figure 1, PCA and CNN are precedented by three other modules that can be presented as follows:

- Dataset Acquisition: like any other classifier, this model requires creating a dataset of tweets that will be used to train our classifier. The tweets can be collected from Twitter using different technics such as using Twitter streaming API or the R programming language. Fortunately, the number of good quality and publicly available datasets that are relevant to this context still growing. In this work, we opted for the HoneyPot dataset that is created specifically for distinguishing the activities of content polluters (i.e., spammers) and legitimate users on Twitter.
- Preprocessing: after the acquisition step and before the extraction of features, our data need to be cleaned to increase its quality and provide a better input into the classifier. Our preprocessing protocol include conversion to lowercase, the removal of punctuation, stop-words, and white spaces; tokenization, Stemming and Lemmatization.
- Feature Selection: this step has a significant impact on the overall performance of the proposed classifier. In recent literature, most of the basic features that have been identified and exploited in Twitter spam detection tasks can be categorized into five principal classes, namely, content-based, user-based, and hybrid techniques.

The main contribution of this paper is that we put an extra layer just after the embedding layer (i.e., the former input layer of CNN) that will generate using PCA a new reduced and most pertinent collection of embeddings that will serve as the new input layer for the traditional CNN model [21-23].

To generate the first embeddings, we opted for Google's word2vec [24-26] that converts the preprocessed tweets into a collection of numerical vectors. Hence, each tweet will be converted into a matrix so that the rows represent the words and the columns represent the embeddings. Usually, word2vec uses a high-dimensional vector space to learn the embeddings, typically of several hundred dimensions. In our experiments, we used 300-dimension word vectors to train the proposed classifier.

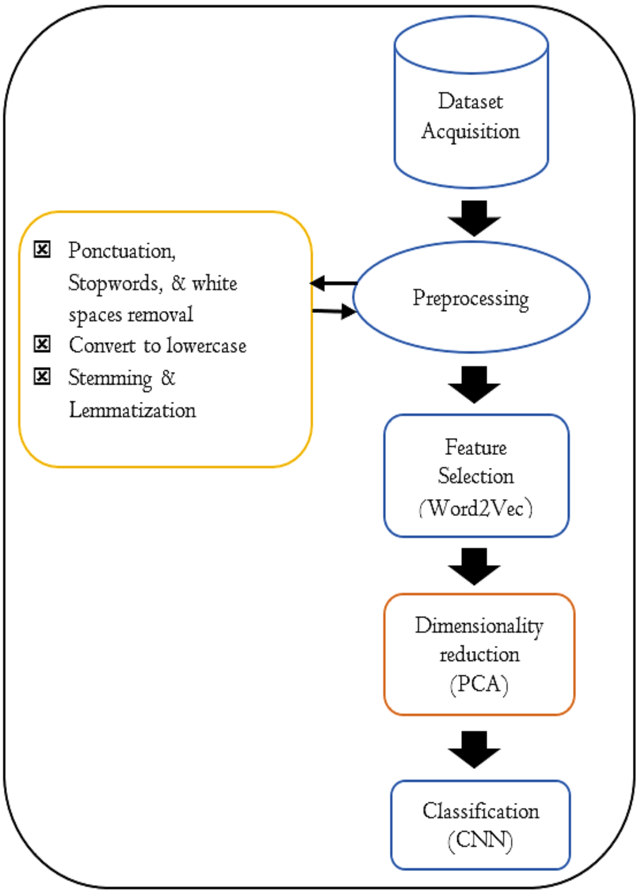
Dataset Acquisition
☒ Ponctuation, Stopwords, & white spaces removal
☒ Convert to lowercase
☒ Stemming & Lemmatization
Preprocessing
Feature Selection (Word2Vec)
Dimensionality reduction (PCA)
Classification (CNN)

Figure 1. An overview of the proposed model

4. EXPERIMENT RESULTS AND EVALUATION

To test the performance of the proposed model, we conducted a series of experiments after its implementation using Keras 2.10 With TensorFlow Backend on Ubuntu 18.04. We have created a dataset that contains around 5200 public conversation tweets using Python and the Twitter API through the developer portal.

As shown in Table 1, all the collected tweets have been manually labelled as either spams or non-spams and divided into 60% vs 40% as training and testing subsets to train and test the classifier.

Table 1. Spam and Non-Spam tweets count

Tweets	Count
Spams	2712
Non-Spams	2488
Total	5200

To train the classifiers, grid search over diffrent combinations of the batch size and epochs suggests that the values that work best for our experiments are set respectively to 15 and 50.

We selected four evaluation metrics to evaluate the classification performance of the proposed model, namely:

- Accuracy: It is described as the ratio of the true predictions to the total number of predictions.

$$Accurracy = \frac{(T^+ + T^-)}{(T^+ + F^+ + F^- + T^-)}$$

- Precision: It represents the fraction of the true positive predictions with respect to the total number of positive predictions.

$$Precision = \frac{T^+}{T^+ + F^+}$$

- Recall: It is the fraction of all the true predictions returned by the classifier.

$$Recall = \frac{T^+}{T^+ + F^-}$$

- F-Score: as a combination of recall and precision, it captures the properties of both measures to summarize the classification performance of the model.

$$\mathrm{F-Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In the above formulas, T^+ , T^- , F^+ , and F^- stand respectively for True Positive, True Negative, False Positive, and False Negative as shown in Table 2.

Table 2. Confusion matrix

	Non-Spam	Spam
Non-Spam	False Negative (F^-)	True Negative (T^-)
Spam	True Positive (T^+)	False Positive (F^+)

Concerning the best choices for the number and size of filters as well as the activation function to train the CNN, we opted for the same values adopted by Sharmin et al. [17], as they observed that 64 4-size filters with ReLU activation function return the best classification results.

In order to investigate the effect of the eigenvectors that are obtained by the application of PCA algorithm over the embedding vectors, we have tested the classifier with different numbers of those vectors.

Let's recall the steps of PCA algorithm then we apply them to our data. we firstly assume that a given tweet that we want to check whether it is a spam or not is in the two-dimensional matrix form (let's call it Γ), with M lines representing the word2vec embeddings and N columns representing the words. In our experiments, we take the number of embeddings m to be 300, which means that each word in the tweet is considered as a point in a 300-dimensional space.

- Step 1: we calculate the mean vector of the matrix Γ (let's call it φ) as follows:

$$\varphi = \frac{1}{N}\sum_{n=1}^{N}\Gamma_n$$

where, Γ_n stands for the word vector whose column is indexed by n.

- Step 2: we subtract the average vector from each column of the matrix Γ, and a new matrix θ is obtained as follows:

$$\theta = \Gamma - \varphi$$

- Step 3: we calculate the covariance matrix C as follows:

$$C = \frac{1}{M}\sum_{n=1}^{M}\theta_n\theta_n^T$$

- Step 4: we calculate the eigenvectors with the corresponding eigenvalues for the new matrix C.

We emphasis that the matrix C has a dimensionality of $N^2 * N^2$, which means that a set of N^2 eigenvectors and eigenvalues will be generated.

However, it should be noted that this number is practically very huge, and most of those eigenvectors are considered irrelevant for the classification process. It is precisely for this reason that PCA algorithm intervenes to reduce the feature space dimensionality of the initial data (i.e., word embeddings) even though it inevitably brings some loss of information [27, 28].

- Step 5: Rather than using a very large $N^2 * N^2$ matrix, PCA suggest using an alternative matrix with a very low dimensionality, namely, $\grave{C} = \theta_n^T\theta_n$.

The dimensions of the matrix $\grave{C}$ is $M * M$, so we would have M eigenvectors corresponding to the M highly ranked eigenvectors of the matrix C.

- Step 6: We take K eigenvectors which correspond to the K largest eigenvalues from the newly generated set of M eigenvectors to shape a new feature space.

- Step 7: Each embedding vectors will be projected into the new feature space to obtain a k-dimensional weight vector. The values of this vector are obtained by the multiplication of the embedding vector minus the mean vector with each eigenvector as follows:

$$\omega_i = E_i^T(\Gamma - \varphi)$$

where, ω_i and E_i^T refer to the i^{th} weight and eigenvector respectively.

The weight vector of the embedding vector Γ that will be fed as the new input in the model is represented as follows:

$$\Psi = [\omega_i, \omega_i, \omega_i, \dots, \dots, \omega_k]$$

Table 3. Performance comparison of the proposed model with the ANN and CNN models

Classifier	Precession	Recall	Accuracy	F-Score
ANN	92.86	91.92	91.87	92.39
CNN	93.71	97.06	93.02	95.36
SVM	94.02	95.87	94.91	94.92
PCA-CNN	94.91	96.76	95.12	95.83

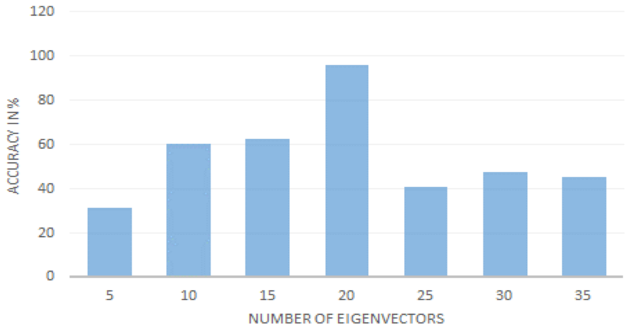

Figure 2. Variation of accuracy with the number of eigenvectors

In the second part of our experiments, the proposed model is compared against three classical benchmark classifiers, the Artificial neural network (ANN) [29, 30], the Convolutional neural network (CNN) [31, 32], and the Support vector machine (SVM) [33, 34].

In Table 3, the values of all the metrics show that the proposed model performs more accurately than the traditional classifiers with precession of 94.91, recall of 96.76, and F-score of 95.83. The values of the different metrics suggest that CNN deep learning classifier performs significantly better than the two machine leaning benchmarks ANN and SVM. It should be mentioned that even if the gap between the values of the different metrics does not seem significant between PCA-CNN and CNN, the later takes a relatively longer training time. A plausible explanation for the findings (Figure 2) is that the stacking of PCA and CNN have reduced the effect of overfitting by preserving the most important data (word embeddings) in the form of highly ranked eigenvectors and by eliminating irrelevant data which is represented in lower-level word embeddings [35-37].

5. CONCLUSION

In this paper, we have introduced a simplistic deep learning classifier for the detection of potential spammers in Twitter by the combination of PCA and CNN. The basic idea behind this new approach is that if we reduce the size of the input vectors (i.e., word embeddings) by the projection into a low-dimensional feature space, we can avoid overfitting, minimize the training time, and increase the classification performance. The results revealed that PCA-CNN outperforms existing benchmarks like ANN, CNN and SVM.

REFERENCES

[1] Loucif, H., Akhrouf, S. (2022). Toward a new recursive model to measure influence in subscription social networks: A case study using Twitter. In International Conference on Managing Business Through Web Analytics, pp. 131-141. https://doi.org/10.1007/978-3-031-06971-0_10

[2] (2022). Number of X (formerly Twitter) users worldwide from 2019 to 2024. Social Media & User-Generated Content, https://www.statista.com/statistics/303681/twitter-users-worldwide/, accessed on June. 17, 2023.

[3] (2023). 19.42% of active Twitter accounts are fake or spam: Analysishttps://mediamakersmeet.com/19-42-of-active-twitter-accounts-are-fake-or-spam-analysis/, accessed on June. 19, 2023.

[4] Loucif. H. (2021). A simplistic model for spammers detection in social recommender systems. International Journal of Business Information Systems, 1(1): 1. https://doi.org/10.1504/IJBIS.2021.10044172

[5] Wu, T., Wen, S., Xiang, Y., Zhou, W. (2018). Twitter spam detection: Survey of new approaches and comparative study. Computers & Security, 76: 265-284. https://doi.org/10.1016/j.cose.2017.11.013.

[6] Inuwa-Dutse, I., Liptrott, M., Korkontzelos, I. (2018). Detection of spam-posting accounts on Twitter. Neurocomputing, 315: 496-511. https://doi.org/10.1016/j.neucom.2018.07.044

[7] Aljabri, M., Zagrouba, R., Shaahid, A., Alnasser, F., Saleh, A., Alomari, D.M. (2023). Machine learning-based social media bot detection: A comprehensive literature review. Social Network Analysis and Mining, 13(1): 20. https://doi.org/10.1007/s13278-022-01020-5

[8] Abdelwahab, A., Mostafa, M. (2022). A deep neural network technique for detecting real-time drifted twitter spam. Applied Sciences, 12(13): 6407. https://doi.org/10.3390/ app12136407

[9] Elmendili, F., Bouzekri El Idrissi, Y.E. (2020). A framework for spam detection in Twitter based on recommendation system. International Journal of Intelligent Engineering & Systems, Ibn Tofail University, Kenitra, Morocco, 13(5): 85-96. https://doi.org/10.22266/ijies2020.1031.09

[10] Jain, G., Sharma, M., Agarwal, B. (2019). Spam detection in social media using convolutional and long short term memory neural network. Annals of Mathematics and Artificial Intelligence, 85(1): 21-44. https://doi.org/10.1007/s10472-018-9612-z

[11] Tai, K.S., Socher, R., Manning, C.D. (2015). Improved semantic representations from tree-structured long short-term memory networks. arXiv Preprint arXiv: 1503.00075. https://doi.org/10.48550/arXiv.1503.00075

[12] Abebaw, Z., Rauber, A., Atnafu, S. (2022). Design and implementation of a multichannel convolutional neural

network for hate speech detection in social networks. Revue d'Intelligence Artificielle, 36(2). https://doi.org/10.18280/ria.360201

[13] Gharge, S., Chavan, M. (2017). An integrated approach for malicious tweets detection using NLP. In 2017 International Conference on Inventive Communication And Computational Technologies (ICICCT), Coimbatore, India, IEEE, Coimbatore, India, pp. 435-438. https://doi.org/10.1109/ICICCT.2017.7975235

[14] Shahariar, G.M., Biswas, S., Omar, F., Shah, F.M., Hassan, S.B. (2019). Spam review detection using deep learning. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, IEEE, pp. 0027-0033. https://doi.org/10.1109/IEMCON.2019.8936148

[15] Fazil, M., Abulaish, M. (2018). A hybrid approach for detecting automated spammers in Twitter. IEEE Transactions on Information Forensics and Security, 13(11): 2707-2719. https://doi.org/10.1109/TIFS.2018.2825958

[16] O'Shea, K., Nash, R. (2015). An introduction to convolutional neural networks. arXiv Preprint arXiv: 1511.08458. https://doi.org/10.48550/arXiv.1511.08458

[17] Sharmin, T., Di Troia, F., Potika, K., Stamp, M. (2020). Convolutional neural networks for image spam detection. Information Security Journal: A Global Perspective, 29(3): 103-117. https://doi.org/10.1080/19393555.2020.1722867

[18] Gewers, F.L., Ferreira, G.R., Arruda, H.F.D., Silva, F.N., Comin, C.H., Amancio, D.R., Costa, L.D.F. (2021). Principal component analysis: A natural approach to data exploration. ACM Computing Surveys (CSUR), 54(4): 1-34. https://doi.org/10.1145/3447755

[19] Greenacre, M., Groenen, P.J., Hastie, T., d'Enza, A.I., Markos, A., Tuzhilina, E. (2022). Principal component analysis. Nature Reviews Methods Primers, 2(1): 100. https://doi.org/10.1038/s43586-022-00184-w

[20] Fei, N.Y., Gao, Y.Z., Lu, Z.W., Xiang, T. (2021). Z-Score Normalization, Hubness, and Few-Shot Learning. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, pp. 142-151. https://doi.org/10.1109/ICCV48922.2021.00021

[21] Alom, Z., Carminati, B., Ferrari, E. (2020). A deep learning model for Twitter spam detection. Online Social Networks and Media, 18: 100079. https://doi.org/10.1016/j.osnem.2020.100079

[22] Ban, X.B., Chen, C., Liu, S.G., Wang, Y., Zhang, J. (2018). Deep-learnt features for Twitter spam detection. International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec), Santa Clara, CA, USA, pp. 208-212. https://doi.org/10.1109/SocialSec.2018.8760377

[23] Santoshi, K.U., Bhavya, S.S., Sri, Y.B., Venkateswarlu, B. (2021). Twitter spam detection using naïve bayes classifier. In 2021 6th international conference on inventive computation technologies (ICICT), Coimbatore, India, pp. 773-777. https://doi.org/10.1109/ICICT50816.2021.9358579

[24] Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv Preprint arXiv: 1301.3781. https://doi.org/10.48550/arXiv.1301.3781

[25] Sivakumar, S., Videla, L.S., Kumar, T.R., Nagaraj, J., Itnal, S., Haritha, D. (2020). Review on word2vec word embedding neural net. In 2020 international conference on smart electronics and communication (ICOSEC), Trichy, India, pp. 282-290. https://doi.org/10.1109/ICOSEC49089.2020.9215319

[26] Mazumder, T., Das, S., Rahman, M.H., Helaly, T., Pias, T.S. (2022). Performance evaluation of different word embedding techniques across machine learning and deep learning models. In 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh pp. 932-937. https://doi.org/10.1109/ICCIT57492.2022.10055572

[27] Yeh, M., Gu, M. (2022). An efficient and reliable tolerance-based algorithm for principal component analysis. 2022 IEEE International Conference on Data Mining Workshops (ICDMW), Orlando, FL, USA, pp. 642-649. https://doi.org/10.1109/ICDMW58026.2022.00088

[28] Naveen, S., Omkar, A., Goyal, J., Gaikwad, R. (2022). Analysis of principal component analysis algorithm for various datasets. In 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, pp. 1-7. https://doi.org/10.1109/INCOFT55651.2022.10094448

[29] Obeidat, M.A., Mansour, A.M., Al Omaireen, B., Abdallah, J., Khazalah, F., Alaqtash, M. (2021). A deep review and analysis of artificial neural network use in power application with further recommendation and future direction. In 2021 12th International Renewable Engineering Conference (IREC), Amman, Jordan, pp. 1-5. https://doi.org/10.1109/IREC51415.2021.9427846

[30] Mishra, M., Srivastava, M. (2014). A view of artificial neural network. In 2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014), Unnao, India, pp. 1-3. https://doi.org/10.1109/ICAETR.2014.7012785

[31] Albawi, S., Mohammed, T.A., Al-Zawi, S. (2017). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, pp. 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[32] Aloysius, N., Geetha, M. (2017). A review on deep convolutional neural networks. In 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, pp. 0588-0592. https://doi.org/10.1109/ICCSP.2017.8286426.

[33] Wang, Q. (2022). Support vector machine algorithm in machine learning. In 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, pp. 750-756. https://doi.org/10.1109/ICAICA54878.2022.9844516

[34] Shrivastava, V., Karsoliya, S., Verma, B., Gupta, N.K. (2021). Social data analysis: Cyber recruitment analysis spam detection over Twitter dataset using SVM & ARIMA model. In 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, pp. 1-7. https://doi.org/10.1109/ICAECT49130.2021.9392543.

[35] Al-Azani, S., El-Alfy, E.S.M. (2018). Detection of arabic spam tweets using word embedding and machine learning. In 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, pp. 1-5.

https://doi.org/10.1109/3ICT.2018.8855747.

[36] Shah, P., Shah, S., Joshi, S. (2022). A study of various word embeddings in deep learning. In 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-5. https://doi.org/10.1109/INCET54531.2022.9824963

[37] Jiao, Q., Zhang, S. (2021). A brief survey of word embedding and its recent development. In 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, pp. 1697-1701. https://doi.org/10.1109/IAEAC50856.2021.9390956