Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Detection of Compromised Online Social Network Account with an Enhanced Knn

Edward Kwadwo Boahen, Wang Changda & Bouya-Moko Brunel Elvire

To cite this article: Edward Kwadwo Boahen, Wang Changda & Bouya-Moko Brunel Elvire (2020) Detection of Compromised Online Social Network Account with an Enhanced Knn, Applied Artificial Intelligence, 34:11, 777-791, DOI: 10.1080/08839514.2020.1782002

To link to this article: https://doi.org/10.1080/08839514.2020.1782002

Published online: 29 Jun 2020.

APPLIED ARTIFICIAL INTELLIGENCE
2020, VOL. 34, NO. 11, 777–791
https://doi.org/10.1080/08839514.2020.1782002

Taylor & Francis
Taylor & Francis Group

Detection of Compromised Online Social Network Account with an Enhanced Knn

Edward Kwadwo Boahen, Wang Changda, and Bouya-Moko Brunel Elvire

Department of Computer Science and Technology, Jiangsu University, Zhenjiang, China

ABSTRACT

The primary threat to online social network (OSN) users is account compromisation. The challenge in detecting a compromised account is due to the trusted relationship established between the account owners, their friends, and the service providers. The available research which focuses on using machine learning has limitations with human experts involved in feature selection and a standardized dataset. The paper discusses users` various behaviors of users of OSN and the up-to-date approaches in detecting a compromised OSN account with emphasis on the limitations and challenges. Furthermore, we propose an enhanced machine learning approach Word Embedding and KNN (WE-KNN), which addresses the limitations faced by the previous techniques used. We detailed our proposed WE-KNN for feature extraction, selection of behavior of OSN users, and classification. Our proposed model is evaluated using the standard benchmark datasets, namely KDD Cup '99 and NSL-KDD and implemented it in WEKA. Besides, we used state-of-the-art evaluation metrics to assess the performance of our model. The results obtained depicts that the proposed approach in compromise account detection performs better.

KEYWORDS

Online Social Network; Compromised Account; WE-KNN

Introduction

While OSNs bring us the extraordinary convenience of interactions with many people, they inevitably come with risk and uncertainty, making the issue of whether to trust those we interact with more significant (Chen et al. 2019). Without trust, the benign development of OSNs would not be possible (Chen et al. 2019). Therefore, monitoring and managing privileged user actions are imperative for cybersecurity and compliance reporting. One of the most relevant and robust techniques is profiling users and creating a user model to monitor and detect anomalies (Lashkari, Chen, and Ghorbani 2019).

Research by (Lashkari, 2019) (Kaushal, 2018) describe a user's profile as a typical pattern that contains the preferences and tendencies of the user. It further stated that knowledge obtains from profiles of users provides a high indication of the users thinking and can be used to predict the user's

CONTACT Wang Changda wang.changda@qq.com Department of Computer Science and Technology, Jiangsu University, Zhenjiang, China

intentions. A user's behavioral tendencies can be practically predicted based on the users existing behavior model (Lashkari, 2019) (Kaushal , 2018). (Kaushal, 2018) All in all, how to effectively and efficiently model social users, have been regarded increasingly as a highly demanded and valuable research challenge (Kaushal, 2018) (Cai et al. 2019). Many profiling evaluation methods have been proposed based on data analytics, machine learning, embedding's, and other techniques. However, most of the proposed methods focus is based on the feature selection that is used for the classification (Kanodia et al, 2018) (Singh et al, 2018) and neglecting data processing, which has an adverse effect on the output of the classification. Research available focuses on feature selection by supervised methods either by the filter, wrapper, or hybrid method. However, much emphasis is given to the method of classification on compromised accounts concerning the profiling of OSN with less emphasis on data processing and feature selection/extraction, which can also affect the output of the classification. Machine learning techniques such as Bayes Net, Logistics Regression, J48, Random Forest, AdaBoostM1, Markov decision approach, and Support Vector Machine have been applied in the area of Profiling OSN for compromised account detection (Kanodia et al, 2018) (Singh et al, 2018)

However, the application of the technique has dramatically improved the detection rate with respect to OSN account compromisation yet with some limitations due to the selection of features, finding suitable data and patterns. Aside it being labor-intensive and expensive, it is also prone to so many errors when the data is large (Zhao et al. 2016).

To improve the effectiveness and accuracy of the anomaly detection model, the analysis will require in-depth monitoring and granularity. Profiling analysis must be more detailed and contextually-aware, hence the need for tuning during the experiment. We propose a new technique known as WE-KNN for feature extraction and Classification to improve the output of classification as compared to existing schemes. This research combines both learning techniques in other to complement each other`s strengths to reduce analytical overheads. Our proposed scheme and tuning can be used to facilitate a more in-depth analysis of a network`s data and OSN account compromisation. Our proposed model performed better than the existing models like SVM, Naive Bayes, Random Forest, and KNN. Compared to existing schemes, our scheme comes with the following advantages.

- Extracting the exact vectors from sentences, characters, and strings hence improving the efficiency of the classification with respect to computational time.
- Tuning is employed to determine which rules or parameters are of much relevance to the current environment in other to improve the performance of the scheme. This helps to reduce the overhead and lowers the chances for FP rate, FN Rate, and CER.

The remainder of this paper is organized as follows, Section II related work. Word Embedding's & tuning is discussed in Section III. Section IV discusses the Datasets. The proposed methodology is discussed in Section V. Discussion on our findings and Limitations. VI Finally, Section VII concludes the paper.

Related Work

Machine Learning Approaches

(Al-Janabi et al., 2017) Focuses on a built classification model using supervised machine learning to detect malicious content distribution in OSNs. The study used features from multiple sources to detect malicious URLs from OSN posts. Twitter API and VirusTotal were used for data collection and data labeling, respectively. The classification model used was a random forest combined with features that were derived from a range of sources. The model had a recall value of 0.89 without any tuning plus feature selection. 0.92 was achieved after applying the tuning and feature selection method. The main focus was to demonstrate systematically a way to analyze the RF application in spam detection and highlight the importance of using the appropriate analysis during the process of setting up the RF parameters. Computationally costly MDA wrapper method was the best method for the feature set reduction but had a relatively close performance. (Ala' et al. 2018) Proposed a model from a hybrid machine learning, which is based on the combination of Support Vector Machine (SVM) and Whale Optimization Algorithm, which is one of the metaheuristic algorithms. The proposed model was used to identify spammers in OSNs. The model performs the detection of spammers automatically and gives adequate insight into the feature with most influence during the process of detection. The WOA model is used to perform a dual simultaneous task, which is the optimization of the SVM and feature selection task, respectively. However, four lingual datasets: Arabic, Spanish, English, and Korea were collected from Twitter and used to test and apply the proposed model. In terms of accuracy, the proposed model performed better than the other algorithms and gave a challenging output concerning the recall, precision, AUC, and f-measure. While helping in the detection process to identify the most influencing feature.

The authors proved that factors with the most influence are content and behavior-based, and followed by characteristic-based features. The increasing number of pornographic pictures on twitter is an indicator loophole on twitters spam detection system. Therefore (Singh et al, 2018) developed an efficient spam detection system using some machine learning techniques such as Bayes Net, Logistic Regression, j48, Random Forest, and AdaBoostM1. Spammers' techniques of attacks evolve; hence some existing techniques do not work with the current trend of spam detection systems.

Therefore, in the work of (Fu et al. 2017), a novel framework was proposed by combining supervised and unsupervised learning for spammer detection by leveraging the temporal evolution patterns of users. The framework is capable of detecting a change in users' activities and also quantify users' evolution patterns.

The above-stated research works by (Al-Janabi et al., 2017) (Ala' et al. 2018) (Singh et al., 2018) (Fu et al. 2017) although had improvement in their works with respect to anomaly classification yet had limitations with data pre-processing hence affecting the final output of their schemes. None of the above research employed tuning to their experiment in other to influence their results. This can also have a negative impact on the output since parameters of such operations are not of an equal level of strength.

Discrete-time Stochastic Control and Probabilistic Graphical Model Approach

The approach in (Kanodia et al, 2018) works with Markov Decision Process to model the problem of detecting YouTube Video spam. Dataset's containing attributes such as View Count, Like Count, Dislike Count, Comment Count, etc. were created, and a sequential decision-making model that utilizes the attributes created was constructed to classify the video as a Spam or legitimate video. The proposed model MDP gave a superior performance as compared to the other models, such as the Decision Tree model, Random Forest, and Ripper methods. A Markov Decision Process approach (MDP) was proposed by (Kanodia et al, 2018) in detecting YouTube video spam. Researchers crawled YouTube by using APIs to extract data, which is made up of videos of different categories, and they manually classified them as spam or legitimate. The data set was then used to create the MDP by formulating the actions, transition probabilities, states, and rewards with respect to the problem being solved. A test set is then used to check the accuracy of the policy returned by the proposed MDP. They tested the dataset with some proposed data mining methods proposed for spam detection for its accuracy. (Ying et al, 2018) Studied user posting behavior at an individual level and employed the hidden Markov model (HDM) to generate users posting sequence. The results from (Ying et al, 2018) proved that classifiers using distance-based clustering method, such as Kmeans, gets even worse performance than using raw posting sequence. This indicates that embedding data into just a one-dimensional feature loses too much information. They also indicated that the simple logistic regression classifier is not effective enough dealing with raw sequential data. Their proposed embedding models, both T-LDA and HMM, gain better performance than the baselines, and the features learned from the first-order HMM clustering model had the highest figure for Area Under the Curve (AUC) and other three metrics as well.

In (Chen et al. 2019), the Authors formalized trust evaluation as a classification problem. The research demonstrated how historical records and profiles of users could be organized into a logical structure based on Bayesian networks in other to identify the users who are trustful without the need to necessary build trust relationships in OSNs. A more detailed feature description represented by hidden variables is what was also considered in their research to identify trustful users. Their results achieve higher values as compared to the other six machine learning methods using Facebook and Twitter datasets. However, structured learning and a significant feature dimension were recommended by their research to improve the model.

The above research by (Chen et al. 2019), (Kanodia et al, 2018) (Ying et al, 2018) did improve their classification with respect to their existing works yet had limitations with factors such as manual classification of data, limited attributes and data pre-processing which affected the results of their research with (Chen et al. 2019) recommending significant feature dimensions and structured learning for future works.

Although a high detection rate and accuracy are achieved in the current studies, nevertheless, the detection schemes used, heavily rely on human experts during feature selection. Also, the existing works have a deficiency of long training times as a result of improper data type representation of data during the data preprocessing phase, which reduces the accuracy of the detection of compromised accounts. Therefore, this work seeks to use Word Embedding and KNN (WE-KNN), and fine-tuning of the parameters of the KNN algorithm to improve the accuracy of detection of the compromised accounts on the OSNs.

Word Embedding & Tuning

Word Embedding

It is a feature learning technique in natural language processing (NLP) which maps words or phrases to vectors of real numbers. From (Ying et al., 2018) (Al-Qurishi et al. 2017) (Shah et al. 2018) (Hazimeh et al., 2019) (Kaur et al., 2018) (Sahoo and Gupta 2019) word embedding's have emerged as one of the powerful tool used to encode relationships between words and bridging the vocabulary gap. It has also led to an enhancement in the work of natural language processing (NLP) (Hazimeh et al., 2019) (Sahoo and Gupta 2019) (Wang and Yang 2018). In the NLP community, the most widely used word embeddings are Word2Vec (Mikolov et al. 2013), Glove (Pennington et al., 2014), and FastText (Mikolov et al. 2017). Since the dataset used contains textual components, there is the need to represent the true meaning in a vector format after processing the raw data. The continuous skip-gram model proposed by Mikolov et al. (Karimi et al. 2018) (Bojanowski et al. 2016) is

considered due to its ability to learn from a dense low-dimensional word vector that is good in predicting the surrounding words with a center word (Hazimeh et al., 2019) (Bojanowski et al. 2016).

The Skip-gram model architecture of Word2Vec in Figure 1 below is used to extract word representations that are used to predict the surrounding words of a document or sentence. The main objective of the model given a sequence of training words: w_1, w_2, w_3, …, w_T, is to maximize the average log probability.

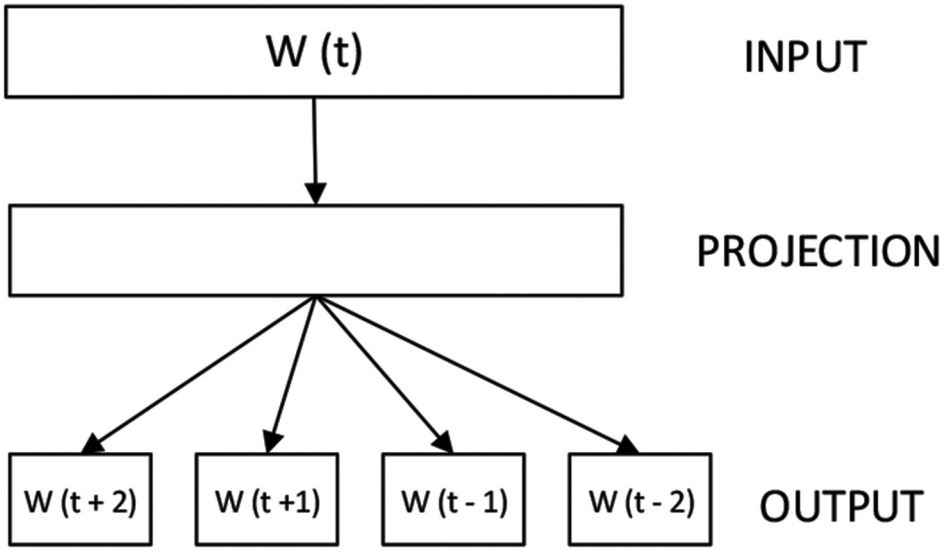

Figure 1. The architecture of the Skip-gram model indicating the output of word representation from the input (W (t)).

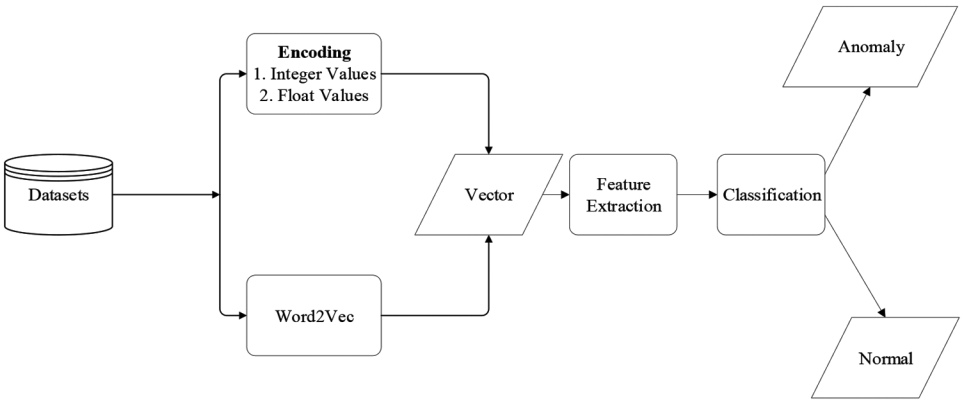

Figure 2. Schematic view of the proposed scheme which starts from data preprocessing to the classification of data.

$$\frac{1}{\mathrm{T}} = \sum_{\mathrm{t}=1}^{\mathrm{T}} \sum_{-\mathrm{c}\leq \mathrm{j}\leq \mathrm{c}, \mathrm{j}\neq 0} \mathrm{logp}(\mathrm{W}_{\mathrm{t+j}}|\mathrm{W}_{\mathrm{t}}) \tag{1}$$

Where c is the size of the training context (this can be a function of the center word w_t). The Larger c results in more training examples, which can lead to higher accuracy, with respect to the time of training. The basic formulae for defining the Skip-gram model is $p(w_{t+j}|w_t)$ using the softmax function:

$$p(W_o|W_1) = \frac{\exp\left(V_{w0}^1 TV_{wI}\right)}{\sum^{w=1} Wexp(V_{w0}^1 TV_{wI})} \tag{2}$$

Where v_w *as "input"* and $v_w^{'}$ as "output" vector representations of w, and W is the number of words in the vocabulary. This formulation is relatively impossible because the cost of computing $\log p(w_O|w_I)$ is proportional to W, which often has a large set of vocabulary (10^5–10^7 terms).

Tuning

Apart from using the parsing of the IOCs to create rule-sets that can be implemented by the IDS, the performance of the IDS is enhanced by tuning some of the parameters of the underlining classifier. This is used to determine which rules are much of relevance to the current environment or process, leading to rules either being disabled or enabled where necessary. Tuning can reduce the overhead and lowers the chances for False Positive rate (FP Rate), false negative rate (FN Rate), and crossover error rate (CER) that is triggered by an unnecessary signature or irrelevant data. Changes do occur in Networks for time, hence the need to implement tuning with recurring intervals. This will ensure the use of a current network profile as the basis for its operating rule-set (Bamler and Mandt 2017).

Datasets

This research utilizes the recognized standard benchmark dataset's in the area of IDS.

Kdd Cup '99

The KDDCUP '99 consists of about 4,900,000 vectors with 41 features, which includes basic features, domain knowledge, and time observation features. The

Table 1. The Standardize Datasets of KDDCUP` 99 and NSL-KDD.

Composition of Dataset					
		10% KDD` 99		NSL KDD	
Category	**Attack Type**	**Train**	**Test**	**Train**	**Test**
	back	2203	1098	956	359
	land	21	9	18	7
	neptune	107201	58001	41214	4657
	pod	264	87	201	41
	smurf	280790	164091	2646	665
Dos	Teardrop	979	12	892	12
	ipsweep	1247	306	3599	141
	nmap	231	84	1493	73
	portsweep	1040	354	2931	157
probe	satan	1589	1633	3633	735
	ftp_write	8	3	8	3
	guess_password	53	4367	53	1231
	imap	12	1	11	1
	multihop	7	18	7	18
	phf	4	2	4	2
	spy	2	0	2	0
	warezclient	1020	0	890	0
R2 L	warezmaster	20	1602	20	944
	loadmodule	9	2	9	2
	Buffer_overflow	30	22	30	20
	rootkit	10	13	10	13
U2 R	perl	3	2	3	2
Normal		97278	60593	67343	9711
Total		494021	292300	125973	18794

Dataset can be downloaded from [http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html]. The vectors are labeled as an attack or normal. From Table 1. Above there are 22 specific types of attack. Research available shows that using 10% of the full KDDCUP dataset is a common practice done to reduce the computational requirement. This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99, the Fifth International Conference on Knowledge Discovery and Data Mining. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

Nsl-Kdd

The NSL-KDD Dataset is the predecessor of the KDD'99 and a refined version (Bojanowski et al. 2016). It has the same structure as the KDDCUP '99 with 22 attack patterns and fields for 41 features. The NSL-KDD dataset contains 41 attributes in each record, unfolds different features of the flow. It also has an assigned label to each of them, which is either an attack or normal. The research takes into consideration the full NSL-KDD dataset for its evaluation

as shown in Table 1 above. The NSL-KDD dataset can be downloaded from [https://www.unb.ca/cic/datasets/nsl.html]

Proposed Methodology

This section gives an overview and a detailed description of each phase of the processes involved in our research. Figure 2 above shows the schematic view of the method used. Below is the schematic view of the method used.

Motivated by the current advancement of word embedding's in the NLP task (Karimi et al. 2018). This research adopts word embeddings to extract word features and transform them into vectors to be utilized by the KNN classifier. The continuous skip-gram proposed by Mikolov et al. (Karimi et al. 2018) is applied. The fundamental idea of using the model is for it to learn dense low-dimensional word vectors that can be good for prediction. The models' output is a dictionary of words, with each associated with a representation of a vector. The result of the description vector and the other values that were encoded are used to train the KNN as an input for the classification. Pre-processing is a requirement on the KDDCUP dataset before it can be successfully implemented with the WE-KNN model. All data that are in phrases or sentences must be pre-processed; likewise, all integer values in other to normalize them since they were combined with floating-point values between "0" and "1" which makes learning difficult and less efficient. This is because the word embedding function will be implemented to convert or transform all the phrases or sentences in the dataset into a vector before the KNN is applied. Another pre-processing will be applied to all integer values in other to normalize them since they were combined with floating-point values between "0" and "1" which makes learning difficult. Below are the main steps of our proposed WE-KNN scheme:

1. Data Pre-processing

- Sorting of Textual data from non-textual data
- Word2Vec is employed to convert strings, characters, and sentences into vectors using the continuous skip-gram (Karimi et al. 2018) before extraction to obtain a dense low- dimensional word vectors for a better classification. (The work and function of skip-gram and word2vec has been discussed above). 2. WE-KNN is then used for feature extraction after data has been processed into normalized vector. 3. WE-KNN is again used to train on the data before classification is done by the use of the Euclidean algorithm which performs i and ii below.
- measure the distance between the new data and all other data that is already classified by using the formula below.

$$d(p,q) = \sqrt{(q1-p1)^2 + (q2-p2)^2 + \ldots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_{i-}p_i)^2} \tag{3}$$

- The smaller distances of K (Parameters) are obtained.

4. Data is then classified as Anomaly or Normal based on the output from the 3rd step.

To use K-Nearest to classify the vector X (where X is an unknown data). It starts by ranking the document`s neighbors among the training datasets (Vectors) and uses the neighbors with the most similar K for prediction of the new data. All the classes of the selected neighbors are weighted and are used to find the similarities of each with respect to X. The similarity is measured by employing Euclidian distance (Cosine value that exists between two document vectors). Below is how it is defined.

$$\mathrm{Sim}\left(\mathrm{X}, D_j\right) = \frac{\left(\sum^{t_i} \in (X \cap^D) \wedge X_i X d_{ij}\right)}{||X||_2 X||D_j||_2} \tag{4}$$

where X is the test data; Dj is the jth training document; t_i is a data shared by X and D_j; x_i represents the weight of data t_i in X; d_{ij} also represents the weight of data t_i in document D_j;

$$||X|| = \sqrt{x_1^2 + x_2^2 + x_3^2 + \ldots} \tag{5}$$

represents the norm of X and ||Dj|| represents the norm Dj. To assign the new document to a known class, there is the need for a cut off threshold.

As compared to other algorithms like Bayesian classifier that rely on prior probabilities, KNN employs similarities of vector instances that are nearby vector space to assume classification. Computationally it is very efficient.

Earlier discussions above prove that the WE-KNN algorithm will perform better in the detection of intrusion with respect to network flow. The figure illustrates the application of the WE-KNN scheme to Intrusion detection. The dataset used contains features such as connection duration, number of packets passed with respect to time, etc. Intrusion detection has only two classes available: “0” for normal and “1” abnormal traffic. However, the computational cost relies hugely on the dimensionality of the vectors and the scale of the training dataset. This is because calculation of the distance between the nodes is done with regards to the dimensions of the vectors involved. To

rectify this anomaly, WE-KNN employs data pre-processing and feature extraction in a way to obtain quality data for comparison. Our method is proven by the results of the experiment discussed in the next session. Traditionally every algorithm performs classification of network traffic into two classes (normal and abnormal) but the advantages of WE-KNN go beyond classification whereby reducing the cost of computation with respect to time-based on the analysis discussed above.

WE-KNN ALGORITHM

```
Let k be the nearest neighbors to be used, m be the size of training dataset, c be the available classes, "0" for
normal and "1" for abnormal, Let r be (instance to be determined)
Input K, m, c, r
for i = 1 to m do
{
Calculate Dyi and D-yi according to definition (3) for each one in training dataset and store;
end for
}
for j = 1 to c do
{
for each instance t in class j
if (Djtk > dist (t,r))
for each instance t not in class j
if (D-jtk > dist (t, r))
end if
end for
end if
end for
}
end for
calculate the p-value of r for class j
classify instance r as class with the corresponding largest p-value, and with confidence (1- the second largest
p-value) and return
Output the largest p-value
```

WE-KNN scheme undergoes three phases which include Data collection and preprocessing, Feature extraction and an intrusion detection phase.

- Data Preprocessing: With our experiment, we employ the NSL-KDD dataset. The dataset is made up of sentences, characters, strings, integers, and numerical data. The word embedding feature is used to extract the exact vectors from the datasets before the vectors are computed. This enhances the quality of the data to be used for the experiment.
- Feature extraction is then applied to the normal data to extract the best feature to be used during training by the algorithm for the detection. Here KNN is employed.
- The last step is the detection phase, where the WE-KNN scheme is introduced to calculate for the *P*-value for each of the instances in the trained dataset discussed above in step 1.

Experiment Results

All experiment was performed using the supercomputer of Jiangsu University. We employed an open-source machine learning framework known as Weka (Mikolov et al. 2013) (Weka 3.9 is the latest version for windows operating system). Weka contains machine learning schemes for data mining activities. Application of algorithm can be applied directly or called using your own java code to a dataset. The application is equipped with tools used for pre-processing of data, classification of data, regression, etc. Machine learning schemes such as Random Forest, SVM, Naive Bayes are all included in the Weka application. We validate our proposed scheme by comparing it to the schemes stated above in Weka. Both the KDD CUP '99 and NSL-KDD datasets were used to perform our evaluation. They are considered as the standard benchmark in the area of intrusion detection concerning research. With this dataset, it is easy to draw a comparison with existing research methods. The metrics below are what we will be using throughout the research.

The following measures were used to evaluate the performance of our model.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{6}$$

Accuracy: The measurement of the proportion of the total number of accurate classifications.

$$Precision = \frac{TP}{(TP + FP)} \tag{7}$$

Precision: This measures the number of accurate classifications penalized by the number of inaccurate classifications.

$$Recall = \frac{TP}{(TP + FN)} \tag{8}$$

Recall: This measures the number of accurate classifications penalized by the number of missed entries.

$$FalseAlarm = \frac{FP}{(FP + TN)} \tag{9}$$

A false alarm is the measure of the proportion of the benign events that are incorrectly classified as being malicious.

The results obtained from our experiments indicate that, in all instances, our model outperformed the existing ones such as SVM, Naïve Bayes, Random Forest, and KNN.

[NB: RESULTS WOULD BE PLACED HERE]

From the results of the experiment above, it is obvious that our proposed We-KNN algorithm out-performed all the state-of-the-art IDS schemes. The false-positive of our proposed model is the lowest as compared to the existing models as indicated in the results from the experiment. Our proposed model requires only the nearest neighbors during the computation of the distance. Tuning was also employed to the parameters to enhance the performance of the detection.

Conclusion

This research proposes a new model built on Word embeddings, and KNN called WE-KNN to improve the accuracy of profiling in an online social network. This approach takes into consideration datasets that have both text, integers, and numbers. We are able to do accurate and better feature selection by employing word embeddings feature in the WE-KNN to achieve higher and better accuracy of 99.98% for recall and 99.97% for precision. We further compare our results from the WE-KNN model against the results obtained by other similar approaches. The results shown above are much improved as compared to previous works done by (Al-Janabi, de Quincey, and Andras 2017) (Ala' et al. 2018) (Singh et al, 2018). We further recommend that a more in-depth analysis be done in the area whiles focusing on optimization to improve the accuracy rate of the subject.

Funding

This work was supported by the National Science Foundation of China under grant 61672269, the National Key Research Project under grant 2017YFB1400703.

References

Ala', A.-Z., H. Faris, J. Alqatawna, and M. A. Hassonah. 2018. Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts. *Knowledge-Based Systems*. doi:10.1016/j.knosys.2018.04.025.

Al-Janabi, M., E. de Quincey, and P. Andras, 2017. Using supervised machine learning algorithms to detect suspicious URLs in online social networks, proceedings of 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Sydney, Australia. doi:10.1145/3110025.3116201.

Al-Qurishi, M., S. Alhuzami, M. AlRubaian, H. MS, A. Alamri, and M. A. Rahman. 2017. User profiling for big social media data using standing ovation model. *Multimedia Tools and Applications*. doi:10.1007/s11042-017-5402-6.

Bamler, R., and S. Mandt. Dynamic word embeddings. In ICML, 380–89,2017. https://arxiv.org/abs/1702.08359

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov. 2016. Enriching word vectors with subword information. doi:10.1162/tacl_a_00051.

Cai, C., L. Linjing, D. Zeng, and H. Ma,2019, Exploring writing pattern with pop culture ingredients for social user modeling, 2019 International Joint Conference on Neural Networks, Budapest, Hungary. DOI: 10.1109/IJCNN.2019.8852187

Chen, X., Y. Yuan, E. Mehm, and A. Orgun. 2019. Using Bayesian networks with hidden variables for identifying trustworthy users in social networks. *Journal of Information Science* pp.1–16. doi:10.1177/0165551519857590..

Hazimeh, H., E. Mugellini, and O. A. Khaled. 2019. Reliable user profile analytics and discovery on social networks. In 2019 8th International Conference on Software and Computer Applications (ICSCA '19), February 19–21, 2019, Penang, Malaysia. ACM, New York, NY, USA, 5 doi: 10.1145/3316615.3316642.

Kanodia, S., R. Sasheendran, and V. Pathari. 2018. A novel approach for youtube video spam detection using markov decision process, Proceedings of International Conference on Advances in Computing, Communications, and Informatics (ICACCI), Bangalore. DOI: 10.1109/ICACCI.2018.8554405

Karimi, H., C. VanDam, L. Ye, and J. Tang. 2018. End-to-end compromised account detection," 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, pp.314–21, doi: 10.1109/ASONAM.2018.8508296.

Kaur, R., S. Singh, and H. Kumar. 2018. AuthCom: authorship verifi- cation and compromised account detection in online social networks using AHP-TOPSIS embedded profiling based technique. *Expert Systems with Applications* 113:397–414. doi:10.1016/j.eswa.2018.07.011.

Kaushal, V., and M. Patwardhan. 2018. Emerging trends in personality identification using online social networks—A literature survey. ACMTrans. Knowl. Discov. Data. 12, 2, Article 15 (January,2018), 30. doi: 10.1145/3070645

Lashkari, A. H., M. Chen, and A. A. Ghorbani. 2019. A survey on user profiling model for anomaly detection in cyberspace. *Journal of Cyber Security and Mobility* 8 (1):75–112. doi:10.13052/jcsm2245-1439.814.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, 3111–19, 2013. arXiv:1310.4546 [cs. CL]

Mikolov, T., E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin. 2017. Advances in pre-training distributed word representations.

Pennington, J., R. Socher, and C. D. Manning. 2014. GloVe: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. 1532–43.DOI:10.3115/v1/D14-1162. https://www.aclweb.org/anthology/D14-1162

Q Fu, B. Feng, D. Guo, and L. Qiang. 2017. Combating the evolving spammers in online social networks. *Computers & Security*. doi:10.1016/j.cose.2017.08.014.

Sahoo, S. R., and B. B. Gupta. 2019. Hybrid approach for detection of malicious profiles in twitter. *Computers and Electrical Engineering* 76:65–81. doi:10.1016/j.compeleceng.2019.03.003.

Shah, S., B. Shah, A. Amin, F. Al-Obeidat, F. Chow, F. J. Moreira, and S. Anwar. 2018. Compromised user credentials detection in a digital enterprise using behavioral analytics. *Future Generation Computer Systems*. doi:10.1016/j.future.2018.09.064.

Singh, M., D. Bansal, and S. Sofat. 2018. Who is who on Twitter–Spammer, fake or compromised account? *A Tool to Reveal True Identity in Real-Time, Cybernetics, and Systems* 49 (1):1–25. doi:10.1080/01969722.2017.1412866.

Wang, C., and B. Yang. 2018 Composite Behavioral Modeling for Identity Theft Detection in Online Social Networks. Social and Information Networks, Cryptography and Security (cs. CR). arXiv:1801.06825v1 [cs.SI]

Ying, Q. F., D. M. Chiu, S. Venkatramanan, and X. Zhang. 2018. User modeling and usage profiling based on temporal posting behavior in OSNs. *Online Social Networks and Media* 8 (2018):32–41. doi:10.1016/j.osnem.2018.10.003.

Zhao, R., R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao. 2016. Deep Learning and its applications to machine health monitoring: A survey, Submitted to IEEE Trans. Neural Netw. Learn. Syst.. [Online]. http://arxiv.org/abs/1612.07640.