Semantic Detection of Targeted Attacks Using DOC2VEC Embedding

Mariam S. El-Rahmany, Ensaf Hussein Mohamed, and Mohamed H. Haggag

Original scientific article

***Abstract*—The targeted attack is one of the social engineering attacks. The detection of this type of attack is considered a challenge as it depends on semantic extraction of the intent of the attacker. However, previous research has primarily relies on the Natural Language Processing or Word Embedding techniques that lack the context of the attacker's text message. Based on Sentence Embedding and machine learning approaches, this paper introduces a model for semantic detection of targeted attacks. This model has the advantage of encoding relevant information, which helps to improve the performance of the multi-class classification process. Messages will be categorized based on the type of security rule that the attacker has violated. The suggested model was tested using a dialogue dataset taken from phone calls, which was manually categorized into four categories. The text is pre-processed using natural language processing techniques, and the semantic features are extracted as Sentence Embedding vectors that are augmented with security policy sentences. Machine Learning algorithms are applied to classify text messages. The experimental results show that sentence embeddings with doc2vec achieved high prediction accuracy 96.8%. So, it outperformed the method applied to the same dialog dataset.**

***Index Terms*—Doc2vec, Multi-class text classification, Pretexting, Sentence Embedding, Targeted Attacks Detection.**

I. INTRODUCTION

THE online social networks presented a communication environment to society and business in general. However, this environment also brings with them different kinds of risks such as cyber-attacks, loss of intellectual property, etc. A Targeted attack is one of the most dangerous types of attacks on a company [1]. It is possible to do so not only through online social networks, but also through text messages in general. This type of attack is a threat that can harm not only the organization's reputation but also it costs a lot. The desire to steal or destroy valuable information drives the majority of cybercrime operations. Targeted attacks may employ well-known online threat methods such as malicious emails, malicious websites, exploits, and malware. The attackers improve and customize their methods based on the nature of their target sector, as well as to circumvent any security measures put in place to reach their target.

Cyber attackers take advantage of the fact that employees are frequently more attackable than computer systems. The attacker persuades the victim to perform a critical action or to elicit critical information. The data collected may contain explicitly valuable information, such as the security information of a bank account, or it may appear to be innocuous, but it may aid the attacker in his attack. An attacker may also persuade the victim to carry out tasks that are considered malicious action, such as restarting a server. So, understanding the intent of the attacker and which security rule he tries to violate is a challenge [2]. Several studies have shown that a system to detect Social Engineering (SE) attacks is required [3]. The detection of targeted attacks from text messages is based on techniques from various fields, most notably Natural Language Processing (NLP) and Machine Learning (ML) methods. However, there are different methods for semantically classifying text messages like statistical methods such as the TF-IDF and Paragraph/Sentence Embedding techniques. The Sentence Embeddings can be performed using various models such InferSent, or Doc2Vec. One hypothesis is that using a variety of semantic relationships between sentences can improve multi-class text classification.

Using a word embedding model to encode all the words of a given sentence and taking the average of all the resulting vectors is a simple and straightforward baseline method for creating sentence vectors. While this provides a solid foundation, it falls short of capturing information about word order and other aspects of overall sentence semantics.
Sentence/paragraph embeddings can be used in nearly all NLP tasks and can significantly outperform counts-based vectorization methods. Sentence embeddings can be adapted for tasks such as semantic search, text clustering, intent detection, and paraphrase detection.

We expected that include security policy vocabulary in the classification model's training set would result in better semantic identification of targeted attacks and restricted action requests. In this paper, the proposed model uses

Manuscript received July 5, 2021; revised October 9, 2021. Date of publication November 24, 2021. Date of current version November 24, 2021. The associate editor prof. Toni Perković has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Department of Computer Science, Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt.

E-mails: Mariam.sk10@gmail.com, ensaf_hussein@fci.helwan.edu.eg, mohamed.haggag@fci.helwan.edu.eg.

Digital Object Identifier (DOI): 10.24138/jcomss-2021-0113

Sentence/paragraph embedding to detect the targeted attack as one of the social engineering attacks. This model provides semantically richer representations. Our goal is to semantically perform multi-class classification for text messages. The organizations should have a security policy to define the security rules that employees should respect to avoid attacks. So, we detect the targeted attack and classify the text message according to the violation of security rules in the security policy.

The main contributions of this paper are summarized as follows:

- The original dataset [6] of text messages was classified as attack (1) or not attack (0) but we reclassified it to one of the four classes. This classification is based on the action that the attacker was attempting to persuade the victim to take.
- A model for detecting social engineering attacks (Targeted attacks) in any text messaging environment is proposed.
- Train the model with the security policy to enrich the model with more vocabularies related to the security domain.
- Use Doc2vec and InferSent as sentence/paragraph embedding techniques to classify text messages as an attack (4 classes) or clean.

The following is how this paper is structured: Section II discusses related work that presents various techniques for detecting targeted attacks. Section III describes our proposed model, techniques to be used, and the datasets that will be used. Section IV discusses classification algorithms. Section V presents the experimental work and results. Finally, in section VI, there is a conclusion and future work.

II. Related Work

Semantic detection of targeted attacks has a great research focus recently. Existing researchers perform this task depending on two steps: extracting semantic features followed by the classification process. Extracting semantic features performed using natural language processing (NLP) and embedding vectors. Then Machine Learning (ML) techniques are used for classification as attack or not.

Some researchers [4],[5],[6] have proposed a method that depends on a list of blacklist topics represented into pairs of action and a resource prepared from the security policy. However, the blacklist technique is inefficient if it depends on a comparison between words. Therefore, the semantic features and machine learning approaches have received more attraction from researchers as shown in the next related research.

Ram Bhakta [4] reforms the security policy manually as pre-defined topics blacklist to check the text if it contains a topic from the blacklist a warning message is generated. Each topic represents two elements: an action and a resource. They do tokenization of the text messages and compare words with the topic blacklist. Their approach achieved 100% precision and 88.9% recall. It does not consider the context of the sentences. Similarly, Yuki Sawa [5] uses the same predefined blacklist. Their approach identifies patterns from a parse tree using Stanford parser. If a sentence contains a question or command, the extracted topic (verb/noun) compared to blacklist topics. The system was tested against two corpora. One of them is a set of three social engineering attacks, while the other is Dialogs Corpus. They obtained 100% precision, 60% recall, and no false positives on the first corpus.

Merton Lansley [6] proposed a technique for detecting Social Engineering (SE) attacks in online chat rooms. It extracts features and then labels the dataset with Intent, Spelling, Link, and attack or no attack. The standard dataset has 148 entries, while the compound dataset has 748 entries include tweets from Twitter's customer support. Classification is done using an artificial neural network (MLP) with an accuracy of 92.2% and the ensemble learning strategy Based on a soft voting approach, a Gaussian Nave Bayes classifier, a Decision Tree classifier, and a Random Forest classifier were used with an accuracy of 92.4 %.

Merton Lansley [7] proposed a method to extract the URL from the text then check if it is malicious or not using Web of Trust (WOT) API. Text messages are compared to a blacklist of topics. This blacklist is derived from a dictionary of security policies. The evaluated method demonstrates that a random forest classifier with 79% accuracy outperforms MLP and K nearest neighbor (KNN).

Andrei Queiroz [8] evaluates different language models with classification algorithms for the detection of malicious messages in hacker communications. Three public datasets are used for testing. Datasets labeled as yes (concerning a type of vulnerability), No (clean), or undecided. They use BOW (Bag of Words) and Word Embeddings models (Word2vec and Glove). They discovered that SVM and BOW outperformed SVM with Word2vec and Glove because they were trained on generic and unrelated security sources.

On five hacker forum datasets, Andrei Queiroz [9] evaluates various language models used in the classification of hacker communication posts. Word Embeddings (Word2vec and Glove) and Sentence Embeddings are language models (Sent2vec, InferSent, and SentEncoder). They discovered that Sentence Embeddings improve SVM classification performance when compared to traditional language models. Their model achieved 88% using Sentence Embedding and SVM.

Cho Cho San [10] proposes a classification system for malware detection. They classify 11 malicious families. The system includes a feature extraction algorithm, feature reduction, and representation procedure for identifying and representing the extracted feature attributes. Various classification algorithms are used like Random Forest (RF), K-Nearest Neighbor (KNN), and Decision Table (DT). Their system achieved 95.8% accuracy. The dataset has a total of 9068 samples was collected from virus share. Malware families are Adware, Backdoor, Downloader, Dropper, EquationDrug, and Packed.

Rim Chaib and et al. [11] proposed a method to classify medical text docouments with one or more label. They apply their method on Ohsumed medical dataset. It consists of a medical abstract covering 23 different types of cardiovascular disease. They use Doc2vec (DBOW and DM architectures) to generate a feature vector for a document in addition to some

handcrafted features. However extracting features increases the performance. Logistic regression algorithm is used and achieved 92% accuracy with Doc2vec (DBOW).

Sentence/paragraph embedding has paved the way for a new way of representing text semantics. The previous studies make use of the Word Embedding and Sentence Embedding techniques. Their work takes into account binary classification or multi-label classification (an attack, not attack, or undecided). Furthermore, these studies do not consider technical terms related to security policy that may be used in the attacker's dialogue.

In this work, the restricted actions from the security policy are used in the training process to leverage the vocabularies with security-related vocabularies. Also, a multi-class text classification is performed with more specialized classes that precisely classify the attacker's intent depend on the action that he persuades the employee to do. In [11] they use Doc2vec to classify medical data, and the results they obtained inspired us to use Doc2vec in our experiment. Our experiment uses two techniques of sentence/paragraph embeddings (InferSent and Doc2vec).

III. Proposed Model

This section describes the proposed model for targeted attack detection with the experiment using two sentence/paragraph embedding techniques. The text messages encoded using Sentence Embedding techniques. The embedding vectors capture context and semantic relations between sentences, which improves classification results. The messages are then classified using various machine learning algorithms.

The text messages are classified upon the list of restricted actions in the security policy document. The category of the attack represents the sentences of the restricted actions or the intent of the attacker and which part of the security policy he wants to violate.

Through the sentence embedding technique, we capture the context and the semantic relationship among sentences. Fig. 1 shows the proposed model's main architecture.

A. Preparing the Dataset

Table I shows a sample from the list of restricted actions from the security policy documents. This table contains sentences that refer to all actions that are considered as a security policy violation [12]. This table is used to enrich the Sentence Embedding model with security-related vocabulary.

These sentences are categorized into four categories:
1-Request sensitive info
2-installing programs
3-do restricted action
4-change HW configuration.

TABLE I
Results A Sample of Classified Actions from a Security Policy

action	category
enter username	request sensitive info
enter password	request sensitive info
enter secuirty code	request sensitive info
enter ip address	request sensitive info
enter credit card number	request sensitive info
enter secuirty code	request sensitive info
enter default email for bank account	request sensitive info
enter credentials	request sensitive info
enter bank id	request sensitive info
setup application	installing programs
install software	installing programs
change ip address	change hw config
change internet configuration	change hw config
reset password	do restricted action
shutdown server	do restricted action
restart device	do restricted action
transfer fees	do restricted action
restart server	do restricted action

- Request sensitive information: when an attacker requests any sensitive information from the victim. For example, asking the employee to send the credential information or security code.
- Installing programs: when an attacker trying to persuade the victim to install any malicious software. For example, asking the employee to install the software.
- Do restricted action: when an attacker trying to persuade the victim to do any restricted action that harms the organization. For example, asking him to shut down or restart the server.
- Change HW configuration: when an attacker trying to persuade the victim to change the configuration of the system. For example, asking him to change the IP address.

The proposed model evaluated on two datasets: Dataset 1 is a dataset proposed in [6] that contains 148 text messages, and Dataset 2 is a publicly available dataset on the UCI ML repository (spam dataset) [13] contains 1000 text messages. Datasets are manually labeled into one of four categories in addition to "Clean": Request sensitive info, Installing programs, Do restricted action, Change HW configuration.

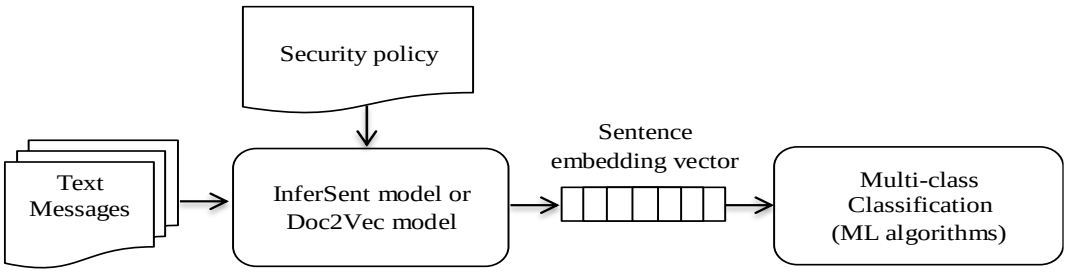

Fig. 1. The main architecture of the proposed model.

B. Preprocessing

The first step in our proposed model is the pre-processing of text messages. Converting text to the lower case, removing stop words, and removing all non-alphabetical characters are all part of this process.

C. Sentence Embedding

Sentence Embedding can be accomplished using a variety of techniques, including InferSent (supervised model) and Doc2vec (unsupervised model) models.
Sentence Embeddings are vectors that represent complete sentences as well as their semantic information. This assists the machine in comprehending the text message's context. We evaluated the two datasets (Dataset 1: 148 entries and Dataset 2: 1148 entries) on the following proposed Sentence Embedding models.

C.1 Doc2vec Embedding

The Doc2Vec embedding is an extension of Word2Vec that adds another "paragraph vector" to the Word2Vec model [14].
Word2vec was proposed as an efficient neural approach to learning high-quality word embeddings. Negative sampling was later introduced as an alternative to the more complex hierarchical softmax step at the output layer, it is more efficient and produces better word vectors on average [15].
The objective function of word2vec is to maximize the log probability:

$$\frac{1}{T}\sum_{i=k}^{T-k} \log\, p\,(w_i \mid w_{t-k}\,, \dots, w_{i+k})$$

where $w_1\,, \dots, w_T$ is a sequence of training words.

A document typically contains hundreds or thousands of distinct words that are considered features; however, many of them may be noisy, less informative, or redundant in relation to class label. This may cause the classifiers to be misled and, as a result, their overall performance to suffer. As a result, feature selection must be used to eliminate noisy, less informative, and redundant features, reducing the feature space to a manageable level and improving the efficiency and accuracy of the classifiers used [16].

Feature extraction is important in text classification because it directly affects classification accuracy. It entails extracting a list of words from text data and then transforming them into a set of features that can be used by a classifier [17].

There are two types of vector representation of words techniques:

- Traditional approaches such as: bag of words and TF-IDF.
- Word embedding based approaches such as: Glove, word2vec, doc2vec, and ELMO.

Sentence embeddings are an extension of the key concepts underlying word embeddings. They broaden the scope of NLP research by representing longer chunks of text as numerical vectors. They share the same fundamental properties as word embeddings, such as the ability to capture a variety of semantic relationships between sentences, such as similarity, contradiction, and entailment.

Fig. 2 shows that D represents the features representing the

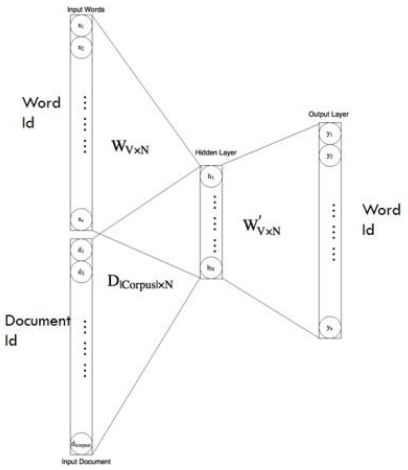

Fig. 2. Paragraph2vector [18]

document context and W represents the word context in a window surrounding the target word. Training is similar to word2vec, with additional document context.

The objective of doc2vec learning is:

$$\max \sum_{\forall(tar,con,doc)} \log P\ (target\ word|context\ words, document\ context).$$

At the end of the training process, you will have word embeddings, W and document embedding D for documents in the training corpus.

So, the proposed model is to add more vocabulary words that are related to the security policy of an organization to improve the contextual prediction.
In the doc2vec architecture, the corresponding algorithms are distributed memory (DM) and distributed Bag of Words (DBOW) [18].

Fig. 3 depicts the DBOW model, which is analogous to the Skip-gram model in word2vec. The document vectors were obtained by training a neural network on predicted words. This model ignores the context words in the input while forcing the model to predict words randomly sampled from the paragraph.

Distributed Memory (PVDM), which is similar to Word2Vec CBOW, is depicted in Fig. 4. A neural network is trained to generate the doc-vectors. This model was trained to predict a center word by averaging context word vectors and the document's doc-vector.

C.2 InferSent Embedding

InferSent [19] is a supervised sentence encoding technique developed by Facebook. It generates semantic sentence representations. However, it uses GloVe or Fasttext vectors for pre-trained word embeddings. The encoder model was trained on supervised data from the Stanford Natural Language Inference datasets. The trained model is based on bidirectional LSTM architecture with max-pooling (SNLI). It helps understand semantic relationships within sentences such that it helps to build a good embedding for sentences. To generate the actual Sentence Embeddings, it takes a pair of sentences

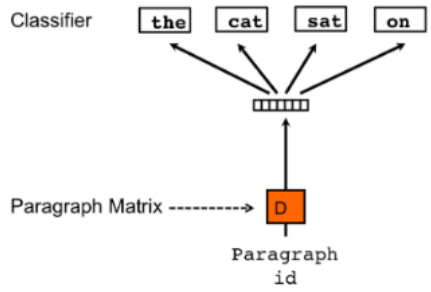

Fig. 3. PVDBOW (The Distributed BOW version of Paragraph Vector) [18]

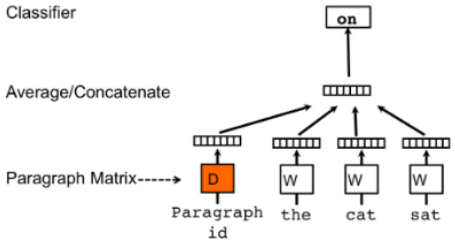

Fig. 4. PVDM (The Distributed Memory version of Paragraph Vector) [18]

and encodes them. Then, as shown in Fig. 5, extract the relationships between these embeddings using the concatenation, element-wise product, and absolute element-wise difference.

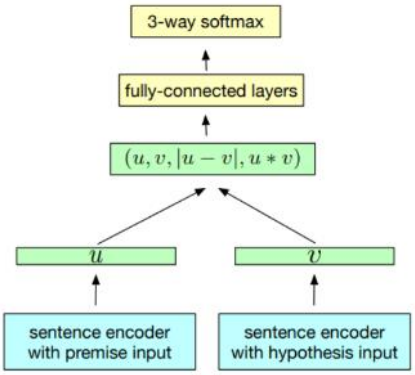

Fig. 5. InferSent model [19]

Glove (Global Vector) [20] is an unsupervised learning algorithm. It introduces a Word Embedding model that combines features from two major word vectors. It provides the semantic relationship between words using the co-occurrence matrix. Words are represented as vectors.

Fasttext [20], a pre-trained word vector model that can be used to train the InferSent model. This model was trained using Common Crawl and Wikipedia. CBOW was used to train these models. In this study, we used dimension 300, character n-grams of length 5, a window of size 5, and 10 negatives.

In our model, the InferSent is set to use the Bi-LSTM with Max pooling architecture as shown in Fig. 6: It is a bi-directional LSTM network that computes n-vectors for n-words, with each vector being a concatenation of output from a forward LSTM and a backward LSTM that read the sentence in opposite directions. Then, to form the fixed-length final vector, a max/mean pool is applied to each of the concatenated vectors.

D. Vectors Generation

Sentence/paragraph embeddings produce sentence representation in numerical semantic vectors. These vectors are used as input to the machine learning model. To generate embedding vectors, we use sentence embeddings to capture the context and semantics of sentences in a vector space model. This is because when Sentence Embeddings are used as the underlying input representation, they have been shown to have a significant impact on the classification task. We

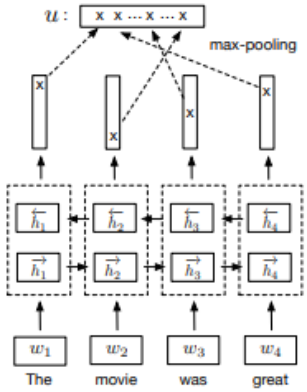

Fig. 6. Bi-LSTM with Max-pooling network [19]

instantiate both models with 300-dimensional feature vectors.

In InferSent, we use GloVe and Fasttext word embedding model. In the Doc2Vec model, we trained the model for 10 epochs, which is the standard number of epochs. The feature vector size is 300.

IV. Classification Phase

This section describes the proposed model for targeted attack detection with the classification model uses the vectors generated in the previous phase as input. The data set is divided into two parts: training and testing. According to studies, when 20-30% of the data is used for testing and the remaining 70-80% of the data is used for training, the best results are obtained [22]. As a result, this model is responsible for training a classifier with 70% of the labeled training dataset and testing the trained classifier's performance with 30% of the testing dataset. We used ML classification models such as Logistic Regression, Random Forest, Nave-Bayes, Linear SVC, and K-Neighbors. These models are used to perform the classification task in both the training and testing phases of our text classification experiment.

During our test: the most similar word to "username" is "credentials" and "send". So, the model shows that the sentence "enter your username and password" is similar to "enter your credentials".

V. Results and Discussion

Comparing our results to the results of the other baseline classification model [6] and other related work [9-11]. Our proposed model shows improvement in classification accuracy as shown in table 6.

The experimental results obtained after implementing our proposed multi-class classification model using Sentence Embedding models are shown in Tables 1 and 2.

A. Performance Metrics

The three metrics Accuracy, Precision, and Recall are used to evaluate the performance. The classification model was evaluated using the Accuracy metric.

This metric is calculated as a ratio of correctly predicted observations to the total observations. The accuracy is represented by Eq. (1), where TP stands for true positives, TN stands for true negatives, FP stands for false positives, and FN stands for false negatives. Eq. (2) denotes precision, Eq. (3) denotes recall, and Eq. (4) denotes the F1 measure.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision is the percentage of relevant text messages correctly retrieved by the system.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

The Recall is the proportion of actual positives identified correctly.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Precision and Recall determine the F1 score. It is employed in order to strike a balance between precision and recall.

$$F1 = 2\ X\ \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

B. Experimental Results

The proposed model was applied to the two datasets. The first has 148 text messages, while the second has 1148. All messages are manually categorized with one of 4 categories (1-Request sensitive info, 2-installing programs, 3-do restricted action, and 4-change HW configuration) in addition to some text messages categorized as clean.

The steps can be summarized as follows: The security policy sentences are added to the training section of the data set after the text messages have been preprocessed. The sentence embedding model (Doc2vec or Infersent) is used to create feature vectors, which are subsequently used to train the ML classifier.

In terms of accuracy, our proposed classification model outperforms the baseline model and other related work, as shown in Table II and Table III.

This demonstrates the significant impact of using security policy sentences to construct the vocabulary for the training model, which resulted in improved classification performance in terms of accuracy.

Using our proposed classification model, the percentage increase is 2.6% for the InferSent model and 4.4% for the Doc2vec model. When the two models are compared, the results show that Doc2vec (DBOW) outperforms InferSent in this experiment. We can also see that the number of dataset entries affects the results. In the experiment, various classification algorithms, such as Logistic Regression, Random Forest, Naïve-Bayes, Linear SVC, and K-Neighbors. The experiment shows that the accuracy of Logistic Regression is higher as compared to the other classification algorithms.

TABLE II
RESULTS A SAMPLE OF CLASSIFIED ACTIONS FROM A SECURITY POLICY ACCURACY (%) RESULTS OF OUR CLASSIFICATION MODEL USING INFERSENT (GLOVE AND FASTTEXT) MODEL

Method	Accuracy			
	Dataset 1		Dataset 2	
	Glove	Fasttext	Glove	Fasttext
Logistic Regression	73.3%	73.3%	**95%**	92.5%
Random Forest	53.3%	55.6%	86.7%	81.7%
Naive-Bayes	64.4%	48.9%	87%	75.4%
LinearSVC	60%	60%	92.4%	88.7%
K-Neighbors	64.4%	66.7%	88.4%	86%

TABLE III
ACCURACY (%) RESULTS OF OUR CLASSIFICATION MODEL USING DOC2VEC (DBOW AND DM) MODEL

Method	Accuracy			
	Dataset 1		Dataset 2	
	DBOW	DM	DBOW	DM
Logistic Regression	82.2%	71.1%	**96.8%**	95.7%
Random Forest	66.7%	66.6%	92.5%	93.3%
Naive-Bayes	60%	48.9%	83%	91.9%
LinearSVC	68.9%	68.9%	90%	95.9%
K-Neighbors	60%	60%	92%	92.5%

Table IV shows different performance metrics on Logistic Regression and InferSent model. While table V shows different performance metrics on Logistic Regression and Doc2vec (DBOW) model.

TABLE IV
PRECISION, RECALL, AND F1 FOR THE LOGISTIC REGRESSION CLASSIFICATION MODEL USING THE INFERSENT MODEL

	precision	recall	f1-score	support
clean	0.98	0.99	0.99	277
request sensitive info	0.71	0.89	0.79	28
installing programs	0.92	0.92	0.92	13
do restricted action	0.90	0.69	0.78	13
change hw config	1.00	0.57	0.73	14
accuracy			0.95	345
macro avg	0.90	0.81	0.84	345
weighted avg	0.96	0.95	0.95	345

TABLE V
PRECISION, RECALL, AND F1 MEASURES FOR THE LOGISTIC REGRESSION CLASSIFICATION MODEL USING DOC2VEC (DBOW)

	precision	recall	f1-score	support
clean	0.99	0.99	0.99	261
request sensitive info	0.88	0.93	0.90	30
installing programs	1.00	0.91	0.95	22
do restricted action	0.89	0.84	0.86	19
change hw config	0.86	0.92	0.89	13
accuracy			0.97	345
macro avg	0.92	0.92	0.92	345
weighted avg	0.97	0.97	0.97	345

Fig. 7 depicts the Confusion matrix for Logistic Regression and the InferSent model. We can see that we classified correctly 25 "Request sensitive info" test samples (out of the total of 28 that are in the test set) and missed 3 that were wrongly predicted. 12 test samples of "Installing programs" correctly classified (out of the total of 13 that are in the test set) and missed 1 that was wrongly predicted. 9 test samples of "do restricted action" correctly classified (out of the total of 13 that are in the test set) and missed 4 that were wrongly predicted. 8 test samples of "change HWconfig" correctly classified (out of the total of 14 that are in the test set) and missed 6 that were wrongly predicted.

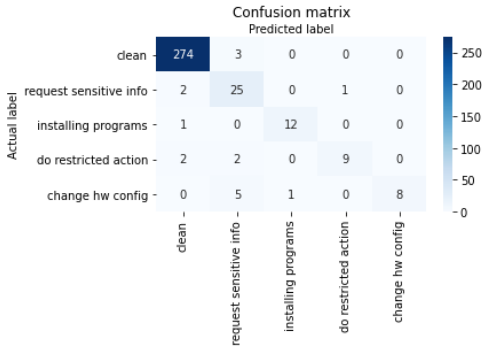

Fig. 7. The confusion matrix for the Logistic Regression classification model using the InferSent (GloVe) model.

Fig. 8 depicts the Confusion matrix for Logistic Regression and the Doc2vec model (DBOW). We can see that we classified correctly 28 "Request sensitive info" test samples (out of the total of 30 that are in the test set) and missed 2 that were wrongly predicted. Twenty test samples of "Installing programs" correctly classified (out of the total of 22 that are in the test set) and missed 2 that were wrongly predicted. 16 test samples of "do restricted action" correctly classified (out of the total of 19 that are in the test set) and missed 3 that were wrongly predicted. 12 test samples of "change HWconfig" correctly classified (out of the total of 13 that are in the test set) and missed 1 that was wrongly predicted.

We used a logistic regression algorithm to achieve 92.17% accuracy with the Doc2vec (DBOW) model without including any sentences from the security policy. During testing, the model with sentences from the security policy achieved 96.8 percent accuracy.

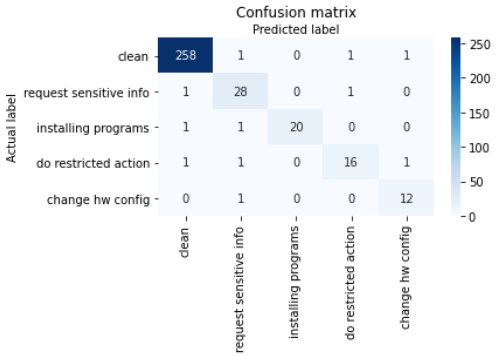

Fig. 8. The confusion matrix for the Logistic Regression classification model using Doc2vec model (DBOW).

In Table VI, we found that this model improved performance when compared to the baselines [6]. This model employs Sentence Embedding in addition to adding the security policy's restricted actions (as sentences). This procedure is carried out during the training process.

TABLE VI
COMPARISON OF ACCURACY (%) RESULTS BETWEEN THE BASE MODEL AND OUR PROPOSED MODEL USING DOC2VEC MODEL (DBOW)

Model	Accuracy
Merton Lansley [6] Ensemble model (Gaussian Naïve Bayes, Decision Tree, Random Forest)	92.4%
Our Model (Logistic Regression, Doc2vec (DBOW))	**96.8%**

VI. CONCLUSION AND FUTURE WORK

A social engineering attack is a dominant and widespread threat to the security of companies. The attacker uses text messages or phone calls to persuade individuals to perform restricted actions. In this paper, we propose a multi-class classification model using the Sentence Embedding techniques and machine learning approach. These techniques provide semantically richer representations. It can generate a far lower-dimensional feature space, including embedding the order and semantics of words, resulting in a more meaningful feature vector. The resulting vectors were used to train our proposed classification model. In the experiment we evaluated two models, InferSent with pre-trained word embeddings and Doc2vec. Embedding vectors enriched with security policy sentences from a company documents that represent security-related vocabulary to improve the classification results. We tested the proposed model using the various machine learning classifiers. The experimental results showed that our proposed model, which used Doc2vec and the Logistic Regression algorithm, outperformed the accuracy results of the baseline dataset used in [6] as well as traditional classification models. However, the proposed model with the Doc2vec outperformed the InferSent model.

Clearly, a larger attention on semantically understanding the context of text messages is required. In the future, we intend to propose a model for detecting all types of social engineering attacks. Semantic features are the primary task for

improving understanding of the attacker's intent. Furthermore, malicious URLs can be detected by classifying the text message rather than just the URL itself.

REFERENCES

[1] N. Tsinganos, G. Sakellariou, P. Fouliras, and I. Mavridis, "Towards an Automated Recognition System for Chat-based Social Engineering Attacks in Enterprise Environments", In: *Proc. of the 13th International Conference on Availability, Reliability, and Security* - ARES 2018. https://doi.org/10.1145/3230833.3233277

[2] M. Hoeschele and M. Rogers, "Detecting social engineering", Detecting Social Engineering. In: Pollitt M., Shenoi S. (eds) Advances in Digital Forensics. DigitalForensics 2005. IFIP — The International Federation for Information Processing, vol 194. Springer, Boston, MA. https://doi.org/10.1007/0-387-31163-7_6

[3] H. Sandouka, A. Cullen, and I. Mann, "Social engineering detection using neural networks", In: Proc. of International Conf. On CyberWorlds, CW '09, pp. 273–278, 2009. https://doi.org/10.1109/CW.2009.59

[4] R. Bhakta and I. G. Harris, "Semantic analysis of dialogs to detect social engineering attacks", In: Proc. of International Conf. On IEEE 9th Int. Conf. Semant. Comput. IEEE ICSC 2015, pp. 424–427, 2015. https://doi.org/10.1109/ICOSC.2015.7050843

[5] Y. Sawa, R. Bhakta, I. G. Harris, and C. Hadnagy, "Detection of Social Engineering Attacks Through Natural Language Processing of Conversations", In: *Proc. of International Conf. On IEEE 10th Int. Conf. Semant. Comput*. ICSC 2016, pp. 262–265, 2016. https://doi.org/10.1109/ICSC.2016.95

[6] M. Lansley, F. Mouton, S. Kapetanakis, and N. Polatidis, "SEADer++: social engineering attack detection in online environments using machine learning", International Journal of Inf. Telecommun., vol. 4, No. 3, pp. 346–362, 2020. https://doi.org/10.1080/24751839.2020.1747001

[7] M. Lansley, S. Kapetanakis, and N. Polatidis, "SEADer++ v2: Detecting Social Engineering Attacks using Natural Language Processing and Machine Learning", In: *Proc. of International Conf. On INISTA 2020 - 2020* Int. Conf. On Innov. Intell. Syst. Appl. Proc.,2020.https://doi.org/10.1109/INISTA49547.2020.9194623

[8] A. Queiroz, S. Mckeever, and B. Keegan, "Eavesdropping hackers: Detecting software vulnerability communication on social media using text mining", In: *Proc. of International Conf. On The Fourth International Conference on Cyber-Technologies and Cyber-Systems*. (pp. 41-48), 2019. https://doi.org/ 10.21427/h5bt-7s13

[9] A. Queiroz, S. Mckeever, and B. Keegan, "Detecting Hacker Threats: Performance of Word and Sentence Embedding Models in Identifying Hacker Communications", In AICS (pp. 116-127), 2019.

[10] C.C. San, M.M.S. Thwin, and N.L. Htun "Malicious Software Family Classification using Machine Learning Multi-class Classifiers.", In: Alfred R., Lim Y., Ibrahim A., Anthony P. (eds) Computational Science and Technology. Lecture Notes in Electrical Engineering, vol 481. Springer, Singapore, 2019. https://doi.org/10.1007/978-981-13-2622-6_41

[11] R. Chaib, N. Azizi, N. Zemmal, D. Schwab, and S.B. Belhaouari, "Improved Multi-label Medical Text Classification Using Features Cooperation.", In: *Saeed F., Mohammed F., Al-Nahari A. (eds) Innovative Systems for Intelligent Health Informatics. IRICT 2020*. Lecture Notes on Data Engineering and Communications Technologies, vol 72. Springer, Cham. https://doi.org/10.1007/978-3-030-70713-2_7

[12] https://attack.mitre.org/

[13] https://archive.ics.uci.edu/ml/index.php

[14] Lau, Jey Han, and Timothy Baldwin. "An empirical evaluation of doc2vec with practical insights into document embedding generation." arXiv preprint arXiv: 1607.05368 (2016). https://doi.org/10.18653/v1/W16-1609

[15] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." arXiv preprint: 1310.4546 (2013).

[16] X. Deng, Y. Li, J. Weng, and J. Zhang. "Feature selection for text classification: A review". Multimed Tools Appl 78, 3797–3816 (2019). https://doi.org/10.1007/s11042-018-6083-5

[17] R. Dzisevič and D. Šešok, "Text Classification using Different Feature Extraction Approaches," 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), 2019, pp. 1-4, https://doi.org/10.1109/eStream.2019.8732167

[18] Q. Le and T. Mikolov. "Distributed representations of sentences and documents." In: Proc. of International conf. on machine learning, pp. 1188-1196. PMLR, 2014.

[19] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," In: *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*. pp. 670-680. *Association for Computational Linguistics, 2017*. https://doi.org/10.18653/v1/D17-1070

[20] J. Pennington, R. Socher, C. Manning: "Glove: Global vectors for word representation." In: EMNLP. pp. 1532-1543, 2014. https://doi.org/10.3115/v1/D14-1162

[21] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, "Advances in Pre-Training Distributed Word Representations", 2017.

[22] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation", 2018.

Mariam S. El-Rahmany is an Assistant Lecturer at The Higher Institute of Optics Technology. She received her MSc in Computer Science from the Helwan University, Egypt, in 2013. She got her BSc in Computer Science from The Higher Institute of Optics Technology, Cairo, Egypt, in 2001. Her research interests include Natural Language Processing, Machine Learning, Deep Learning, and Information Security Applications.

Dr. Ensaf H. Mohamed received her Ph.D. in computer science, faculty of computers and information systems, Helwan University, Cairo, Egypt, 2013. Her recent research focuses on Machine Learning,Deep Learning, Natural Language Processing, Text Mining, and Social Media Analysis. Currently, she is an Associate professor, faculty of computers and information, Helwan University, Cairo, Egypt.

Prof. Mohamed Hassan Haggag is a Professor of Computer Science, Faculty of Computers and Information, Helwan University, Cairo, Egypt. Haggag is the President of the Scientific Account Center, Helwan University, Cairo, Egypt.