ISSN 2278-3091

Volume 9, No.1.5, 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse4991.52020.pdf

https://doi.org/10.30534/ijatcse/2020/4991.52020

Comparative Performance of Machine Learning Methods for Classification on Phishing Attack Detection

Siti Noranisah Wan Ahmad[1], Mohd Arfian Ismail[2], Edi Sutoyo[3], Shahreen Kasim[4], Mohd Saberi Mohamad[5,6]

[1]Faculty of Computing, Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pahang, Malaysia, anisahaniss5132@gmail.com

[2]Faculty of Computing, Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pahang, Malaysia, arfian@ump.edu.my

[3]School of Industrial Engineering, Telkom University, 40257 Bandung, West Java, Indonesia, roja2128@gmail.com

[4]Faculty of Computer Science & Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Johor, Malaysia, shahreen@uthm.edu.my

[5]Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, City Campus, Pengkalan Chepa, 16100 Kota Bharu, Kelantan, Malaysia, saberi@umk.edu.my

[6] Faculty of Bioengineering and Technology, Universiti Malaysia Kelantan, Jeli Campus, Lock Bag 100, 17600, Malaysia, saberi@umk.edu.my

ABSTRACT

The development of computer networks today has increased rapidly. This can be shown based on the trend of every computer user around the world, whereby they need to connect their computer to the Internet. This indicates that the use of Internet is very important, such as for the access to social media accounts, namely Instagram, Facebook, and Twitter. However, with this extensive use, the Internet does not necessarily have the ability to maintain account security in mobile phones or computers. With a low level of security in a network system, it will be convenient for scammers to hack a victim's computer system and retrieve all important information of the victim for their benefit There are many methods that used by scammers to get the important information where phishing attack is the simplest and famous method to be used. Therefore, this study was conducted to develop an anti-phishing method to detect the phishing attack. Machine learning method was proposed as suitable to be used in detecting phishing attacks. In this paper, several machine learning methods were studied and applied in detecting phishing attack. Experiments of the machine learning methods were conducted to investigate which method performed better. Two benchmark datasets were used in the interest to access the ability of the methods in detecting the phishing attack. Then the results were obtained to show the performance of each methods on all dataset.

Key words: Machine learning, Phishing, Classification

1. INTRODUCTION

Phishing attack is a threat to organisations and individuals because scammers intend to steal valuable information. There are several ways that can be used by scammers to steal information. Mostly scammers tend to attack the victim by using an internet browser, email, and short message service. Nowadays, scammers tend to use email to launch their phishing attacks. The phishing attack may succeed as they can manipulate victims using social engineering by sending emails that contain a message to update some information via a fake link. If a victim clicks on the link, the fake link will bring the victim to a fake website, which looks like a legitimate website. At times, in the fake website, the scammer may ask the victim to update information such as login detail, credit card number or bank account details with the interest of stealing valuable information [1]. Currently, phishing attacks constantly growing and becoming a serious problem to the internet users. There are many techniques and methods that have been carried out in to prevent the phishing attack and it is found that using machine learning method is a promising technique to be applied. The implementation of machine learning in detecting phishing attacks can be done because the detection of phishing attacks can be viewed as a classification problem where the attack needs to be labelled as an attack or not.

Machine learning is one of the artificial intelligence applications that enables a system to automatically learn a problem and simultaneously improve the system performance from experience. Presently, machine learning focuses on the construction of a computer system/program where the method enables the system to access data and learn from it. The machine learning method starts with the observation of data to find patterns within the data before making a decision. This process is repeated several times to improve the decision-making process. There are many machine learning techniques, such as Decision Tree Algorithm (DT), K-Nearest Neighbour Algorithm (KNN), Naïve Bayes Algorithm (NB), Random Forest Algorithm (RF), and Support Vector Machine Algorithm (SVM). In this paper, the comparative assessment of the performance of machine learning methods is performed. The machine learning methods that were chosen for this study are DT, KNN, NB, RF, and SVM. The reason behind the chosen methods is because they are widely used in classification problems. The next section, Methods describes the selected machine learning methods used in this study in detail. This is followed by the Methodology section, where it covers the data collection, experiment, and performance measurements applied in this study. Afterwards, the section of Results and Discussion is presented before this paper is concluded in Conclusion.

2. Machine Learning Method

2.1 Decision Tree Algorithm

Decision Tree Algorithm (DT) is an algorithm that belongs to supervised classification algorithms. This algorithm is used in solving regression and classification problems and creating a training model, which will predict a class or value of target variables that are summarised from the training data.

Decision tree can be implemented by using several types of algorithms, including Iterative Dichotomiser 3 (ID3) and C4.5 algorithms. According to [1], ID3 utilises the process for creating a decision tree in the "top-down" form. It has been proven as a very useful method; nevertheless, it still has many constraints. Due to this reason, the algorithm is inapplicable in many real-world situations. The C4.5 algorithm was developed to overcome this problem and has been considered in ID3. Therefore, this study focuses on using C4.5.

The C4.5 algorithm is a successor of the ID3 algorithm. It is a Decision Tree Algorithm used to detect phishing websites that are usually found attached inside spam emails. This algorithm is categorized as a classification algorithm and involves two steps, namely learning step and classification step [2]. According to [3], the algorithm can be expressed in Equations (1), (2), (3), (4), and (5);

$$Info(D) = -\Sigma_{i=1}^{m} p_i \log_2(p_i) \tag{1}$$

$$Info_A(D) = \Sigma_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{2}$$

$$Split\ Info_A(D) = -\Sigma_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{D_j}{D}\right) \tag{3}$$

$$Gain(A) = Info(D) - Info_A(D) \tag{4}$$

$$GainRatio(A) = \frac{Gain(A)}{Split\ Info_A(D)} \tag{5}$$

where *D* is the training set of class-labelled tuples, D_J is a subset of *D*, C_i is denoted as a class-labelled tuple (*i* = 1, ..., m), p_i is the probability of a tuple in D belonging to the class of C_i and |*D*| is the number of tuples in *D*.

According to [2], this algorithm is tested for phishing detection by using Waikato Environment for Knowledge Analysis (WEKA) tools. This test is based on the J48 optimised implementation of C4.5, which will generate a decision tree once the test is completed. The testing dataset used in this study contained 300 websites. Based on the test, it was found that 200 websites were detected as phishing websites. The success and error rates obtained were 0.826 and 0.173, respectively, after the prediction confusion matrix was generated. Therefore, the accuracy of the classifier model that was trained with 750 instances was 82.6%. This algorithm could produce a better result if there was a higher number of rules, which would enable the test dataset to be checked more accurately. Based on this statement, it can be concluded that the higher the number of instances in a training dataset, the more accurate the decision tree is generated.

2.2 K-Nearest Neighbour Algorithm

KNN is a simple algorithm and effective supervised learning method to detect phishing attacks. KNN is based on clustering the input that has the same features. It will decide to place the input in which class category either phishing class or legitimate class. To identify the class category, KNN tests the data input and checks whether the input is near k neighbour. The value of K is relying on the problem and the size of the data. Figure 1 shows the process of KNN classifying an input based on its neighbours [4]. To provide better understanding, according to [4], the black input will be strongly inserted into the legitimate class. This is because the neighbours near the black input area have 3 green inputs and 2 yellow inputs in the black input's dimensional feature space, meaning that the black input has the same features with the green input and is considered as a legitimate class.

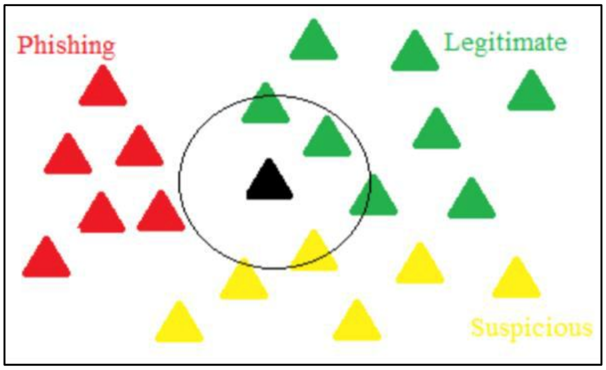

Figure 1 : The process of KNN classifier [4]

KNN finds the nearest input neighbours with the same features from the test data input *e* to train the data input *k* depending on the Euclidean distance to calculate the distance of the neighbours in KNN. For the *k* in the dimensional space for two inputs, $e = [e_1, e_2, \ldots, e_k]$ and $x = [c_1, c_2, \ldots, c_k]$, the two inputs of the Euclidean distance can be expressed in Equation (6) as follows:

$$d\,(e,c) = \sqrt{\sum_{i=1}^{k} (e_i - c_i)^2} \qquad (6)$$

When the KNN is finished with collecting nearest neighbours' input, the majority of the neighbours is treated as a class. Then, the input can be identified as a legitimate link or fake.

2.3 Naïve Bayes Algorithm

NB, which is also known as Bayesian classifier, is a group of common principles, where every characteristic that is being classified is independent of its value among any other characteristics. This algorithm calculates a set of probability based on the combination and frequency of the values [5]. According to [6], the general equation for Bayes' theorem can be expressed in Equation (7) as follows:

$$P(x|Y) = \frac{P(Y|x)P(x)}{P(Y)} \qquad (7)$$

where *P(x)* is the independent probability of *x* (prior probability), *P(Y)* denotes an independent probability of *Y*, *P(Y|x)* represent the conditional probability of *Y* given *h* (likelihood) and *P(x |Y)* gives a conditional probability of *x* given *Y*. According to [7], the NB classifier will use text classification method to filter the spam emails of a victim. It will use tokens, which represent the words used in the spam and non-spam emails, to calculate the probability of the email whether it is a spamming email or not.

The classification of the spam email can be made based on the value of the posterior probability that is obtained. The higher the value of the posterior probability, the more vulnerable the email is, which shows that the probability of the email is a spamming email. The finding in [7] concluded that with the higher number of datasets, the result showed that higher precision and accuracy percentage and smaller percentage of error rate could be obtained. However, the time taken for the experiment to be completed will be longer as the number of datasets increases.

2.4 Random Forest Algorithm

RF is a regression method and learning classification that is suitable for handling data or problems with grouping of data into groups or classes. The RF algorithm works by using decision tree for the prediction concept. According to [8], among other decision tree algorithms, RF performs better in classifying data as it uses forest classification tree to make decisions. In RF, each tree gives a classification, which means the tree vote whether the class is phishing or not. In making the decision, RF chooses the classification that has many votes from the entire tree in the forest. Figure 2 shows how the algorithm classifies the input based on its forest. In the Uniform Resource Locator (URL), there is an instance for RF to classify the data, and then it will make many trees or forests to classify the data. Each leaf will vote either phishing or legitimate; afterwards, the result will come out. In addition, if there are more votes for spam, the result will become spam or phishing data. On the other hand, if there are more votes for not spam, the result will become legitimate data. The determination on deciding on whether the URL is phishing or legitimate can be expressed by the model as in Equation (8) as follows:

$$Gini = \sqrt{1 - \sum_{i=1}^{k} (c_i)^2} \qquad (8)$$

where *i*=1,2, …., *n* and *Gini* indicate that the number of splits across all the tree or root nodes, c_i denoted as probability of particular class while *k* is the number of generated trees.

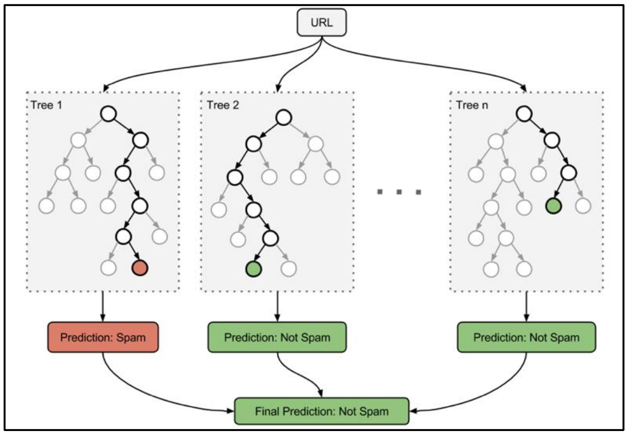

Figure 2 : The process of RF classifier [14]

2.5 Support Vector Machine Algorithm

SVM is used for regression and classification and is related to the supervised learning method. SVM uses a smaller dataset because it takes time and is longer to process [9]. The SVM algorithm will generate a hyperplane to make the best separation of features that have different data elements to ensure that the data is either phishing or legitimate data. The function of the hyperplane is to differentiate between the features. SVM is mapped into the same space and will predict the category based on which side of gap the point or input will fall.

Figure 3 shows the process of SVM. In the figure, the hyperplane is a line of separation between different support vectors. The maximum margin width area is the distance from the surface of the decision to the nearest data point that determines the margins of the classifier. The point that is near and exposed to the margin line is called support vector. The data will be linearly separated when using this algorithm and for this reason, the equation for the algorithm can be expressed in Equations (9) and (10):

$$\min \frac{1}{2} |w|^2 + \sum_{i=1}^{a} \xi_i \tag{9}$$

$$y_i(wx_i - b) \geq 1 - \xi_i \tag{10}$$

where *i* =1, 2, …, *n* and *a* indicate that the number of features, *x* is the number of vector input, *w* is the hyperplane with normal vector, and ξ_i is the parameter to handle non-separable input. When there is a constant occurrence of phishing, this method is suitable to help and know the phishing attack in websites. This SVM method can detect phishing attacks using several features, which are Internet Protocol (IP) address, URL length, shortening service, the @ symbol, double '/' that indicates redirecting, right click, on mouseover, Domain Name System (DNS) record, page rank, Google index, link direct to page, and statistic report [10].

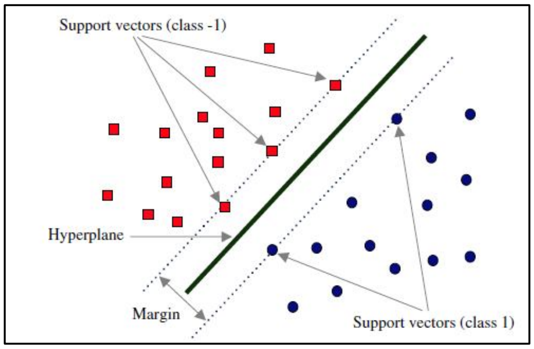

Figure 3 : The process of SVM classifier [16]

3. RESEARCH METHODOLOGY

This section discusses the research methodology involved in carrying out the experiment. The research methodology starts with literature study, data collection, experiment, and performance measurement. The literature study has been presented in the previous section, where several machine learning methods were discussed. The Next subsection is with regard to data collection.

3.1 Data Collection

In this section, the data collection of the study is presented. The datasets that were used in this study were data from email and SMS messages. These datasets consisted of spam and legitimate message. These datasets were used for training purposes (training set).

The first dataset is the email dataset, which can be obtained from the GitHub dataset and was collected by [11]. This dataset can be accessed from https://github.com/waleedalinizami/Spam-Detection-Using-Weka. The dataset contains 5,180 instances and 2 attributes. This dataset features contain a word or character that is frequently occurring in the email. The run-length attributes (55–57) in the email content measure the length of sequences of consecutive capital letters. The format of the dataset is in ARFF type.

The second dataset is the SMS message dataset obtained from the Unicamp website and can be downloaded from http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/. The dataset contains 5,574 instances and 2 attributes. This dataset was collected from various sources, such as Grumbletext Web, NUS SMS Corpus, and SMS Spam Corpus v.0.1 Big. The format of the dataset is in .ARFF file [12].

The datasets used were pre-processed before they were ready to be classified. The process included tokenising. Tokenising is also known as lexical analysis, which involves dividing the content of the text into strings of characters, or tokens. Next, the data are converted from strings to word vectors. Word vector is the vector of numbers that represents the meaning of a word. Simply put, the vector of numbers represents each word of the message content in the dataset. The filtering techniques will be also implemented in this step, where the removal of symbols and white space is performed in this phase. After all the steps of data pre-processing are completed, the dataset is ready to be classified.

3.2 Experiment

Several experiments were performed to measure the performance of machine learning methods. There are many types of cross-validation processes, which include k-fold cross-validation and leave p-out cross-validation. k-fold cross-validation was employed in this study and is the process where the data will be divided into k subsets. Based on the k subsets, each time, one of the k subsets is used as the test set while the other k-1 subsets are put together to form a training set. This will cut down bias as the classification uses most of the data for fitting. Besides, it will also reduce the variances as most of the data are used in the test set. All the results obtained will be recorded and the average of the result will be calculated.

The experiments were run with the coding of Java in the NetBeans IDE software. The WEKA in Java API was used and integrated with the NetBeans software. NetBeans IDE version 8.2 was used in running the experiments. The machine used was Lenovo IdeaPad with processor AMD A9-9420 Radeon powered in Windows 10 Home operating system. For the dataset, the format ARFF was used. When the dataset was successfully read by the software, it would go to the preprocessing data process. In preprocessing, the data were separated by removing the white space and symbols.

3.3 Performance Measurement

After the results were collected in the previous stage, the performance for each machine learning method needed be evaluated. To identify the effectiveness of each method, two metrics classification were used, namely precision and accuracy. Precision is a rate of correct samples that are selected. Meanwhile, accuracy is the measurement of the classification that correctly classifies the category of samples, whether legitimate or phishing. Accuracy and precision can be expressed in Equations (11) and (12):

$$accuracy = \frac{(TN + TP)}{(FP + FN + TN + TP)} \quad (11)$$

$$precision = \frac{TP}{(FN + TP)} \quad (12)$$

where *TN* is true negative (legitimate predicted as legitimate), *TP* is true positive (spam predicted as spam), *FP* is false positive (legitimate predicted as spam), and *FN* is false negative (spam predicted as legitimate).

4. Results and Discussions

In this section, the results obtained from various experiments are presented and compared. The results showed different readings on several aspects, such as the number of correctly classified instances and the time taken for the experiment to be conducted.

The first dataset, which is the email dataset, was conducted and experimented to obtain the result. Table 1 shows the result that has been experimented. From Table 1, KNN, RF, and SVM performed well, where all three algorithms achieved 100% of average accuracy, followed by DT with 98.0502% and NB had the lowest average accuracy with 95.6646%.

Table 1: The full result on first dataset

Methods	Average Accuracy (%)	Average Precision
DT	98.0502	0.981
KNN	100	1
NB	95.6646	0.965
RF	100	1
SVM	100	1

The second dataset, which is the SMS message dataset, was experimented by using all methods. Table 2 shows the result obtained. From the table, the result showed that KNN and RF had better performance than other machine learning methods in terms of detecting phishing attacks. The average accuracy of true instances classified for KNN and RF are 100%, as both methods correctly classified all the data in the dataset. Meanwhile, DT scored the lowest average accuracy, which was 96.78% with 5,395 cases correctly classified in the dataset.

Table 2: The full result on first dataset

Methods	Average Accuracy (%)	Average Precision
DT	96.7887	0.967
KNN	100	1
NB	98.9415	0.989
RF	100	1
SVM	99.9821	0.998

Besides average accuracy, computational time also needed to be taken into account. The comparison of time (in minutes) among all machine learning methods on both datasets is given in Table 3. Based on the table, SVM performed faster as compared to the rest in both datasets. This might be due to the speed of SVM in the classification process since it worked with the maximum margin, which means it allowed very low error in classification. It is a strong and fast model and in solving classification problems [13]. Meanwhile, RF took a longer time in the classification process for all datasets. This is because of the slow process of RF to train the data as it needed to generate many trees, therefore affecting the classification speed [14], [15]. RF is able to handle thousands of data in the dataset at one time for classification; however, the performance will become slow [14].

Table 3: The comparison of time

Methods	Computational time (minutes)	
	First dataset	Second dataset
DT	1.45	0.18
KNN	1.00	0.10
NB	1.23	0.10
RF	1.52	0.20
SVM	0.94	0.07

5. CONCLUSION

Machine learning method is one of the methods that can be utilized in detecting the phishing attack. In this research, five machine learning methods were examined for detecting phishing attacks, which are DT, KNN, NB, RF, and SVM. The used algorithms detected the phishing attacks by classifying the word content in the datasets. Two benchmark datasets were employed, namely email dataset and SMS message dataset. Several experiments were performed to investigate the performance of all selected methods. The performance criteria included the accuracy of correctly classifying the data and the time taken in performing the experiment. From the result, it can be seen that RF has an excellent performance in average accuracy, but it takes a longer time for the classification process. From the results that obtained by the methods, it can be said that even though a method performs well in detecting phishing attacks, it takes a longer time to achieve the best result. Therefore, it is hard to determine which method is better because there is no single method that works well on every problem.

ACKNOWLEDGEMENT

This study was supported by Fundamental Research Grant Scheme (FRGS) with Vot No. FRGS/1/2018/ICT02/UMP/02/2: RDU190113 from Ministry of Higher Education (MOHE) and managed by Universiti Malaysia Pahang.

REFERENCES

1. J. Kozak and U. Boryczka. **Collective data mining in the ant colony decision tree approach,** *Information Sciences*, vol. 372, pp. 126–147, Dec. 2016.
2. A. Priya and E. Meenakshi. **Detection of phishing websites using C4.5 data mining algorithm**, in *IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 2017, pp. 1468–1472.
3. X. YANG, L. YAN, B. YANG, and Y. LI. **Phishing Website Detection Using C4.5 Decision Tree**, in *International Conference on Information Technology and Management Engineering*, 2017, pp. 119–124.
4. A. Taha, **Phishing Websites Classification using Hybrid SVM and KNN Approach**, *International Journal of Advanced Computer Science and Application*s, vol. 8, no. 6, pp. 90–95, 2017.
5. P. Patil, R. Rane, and M. Bhalekar. **Detecting spam and phishing mails using SVM and obfuscation URL detection algorithm**, in *Proceedings of the International Conference on Inventive Systems and Control*, 2017, pp. 1–4.
6. N. Kumar and P. Chaudhary. **Mobile Phishing Detection using Naive Bayesian Algorithm**, *International Journal of Computer Science and Network Security*, vol. 17, no. 7, pp. 142–147, 2017.
7. S.B. Rathod and T. M. Pattewar. **Content based spam detection in email using Bayesian classifier**, in *International Conference on Communication and Signal Processing*, 2015, pp. 1257–1261.
8. V. Muppavarapu, A. Rajendran, and S. Vasudevan. **Phishing Detection using RDF and Random Forests,** *The International Arab Journal of Information Technology*, vol. 15, no. 5, pp. 817–824, 2018.
9. D. K. Srivastava and L. Bhambhu. **Data Classification Using Support Vector Machine**, *Journal of Theoretical and Applied Information Technology*, vol. 12, no. 1, pp. 1–7, 2005.
10. Amrit Kaur. **Detection of Phishing Websites Using SVM Technique,** *Imperial Journal of Interdisciplinary Research*, vol. 2, no. 8, pp. 1273–1276, 2016.
11. V. Metsis, I. Androutsopoulos, and G. Paliouras. **Spam Filtering with Naive Bayes-Which Naive Bayes?**, in *Third Conference on Email and Anti-Spam*, 2006.
12. T. A. Almeida, J. M. Gomez Hidalgo, and T. P. Silva. **Towards SMS Spam Filtering: Results under a New Dataset**, *International Journal of Information Security Science*, vol. 2, no. 1, pp. 1–18, 2012.
13. C.-C. Chang and C.-J. Lin. **LIBSVM: A Library for Support Vector Machines**, *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, May 2011.
14. T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. **How Many Trees in a Random Forest?,** in *Machine Learning and Data Mining in Pattern Recognition* , 2012, pp. 154–168.
15. S. Jagadeesan, A. Chaturvedi, and S. Kumar. **URL Phishing Analysis using Random Forest,** International Journal of Pure and Applied Mathematics, vol. 118, no. 20, pp. 4159–4163, 2018.
16. D. Aksu, A. Abdulwakil, M. A. Aydin, D. Aksu, and M. A. Aydın. **Detecting phishing websites using support vector machine algorithm**, in *World Conference on Technology, Innovation and Entrepreneurship*, 2017, pp. 139–142.