A hybrid method of genetic algorithm and support vector machine for DNS tunneling detection

Fuqdan A. Al-Ibraheemi[1], Sattar Al-Ibraheemi[2], Haleh Amintoosi[3]
[1]College of Dentistry, University of Al-Ameed, Iraq
[2]Education Ministry, Iraq
[3]Faculty of Engineering, Ferdowsi University of Mashhad, Iran

Article Info

Article history:

Received Jan 2, 2020
Revised Jun 29, 2020
Accepted Nov 15, 2020

Keywords:

DNS tunneling
Feature selection
Genetic algorithm
Support vector machine

ABSTRACT

With the expansion of the business over the internet, corporations nowadays are investing numerous amounts of money in the web applications. However, there are different threats could make the corporations vulnerable for potential attacks. One of these threats is harnessing the domain name protocol for passing harmful information, this kind of threats is known as DNS tunneling. As a result, confidential information would be exposed and violated. Several studies have investigated the machine learning in order to propose a detection approach. In their approaches, authors have used different and numerous types of features such as domain length, number of bytes, content, volume of DNS traffic, number of hostnames per domain, geographic location and domain history. Apparently, there is a vital demand to accommodate feature selection task in order to identify the best features. This paper proposes a hybrid method of genetic algorithm feature selection approach with the support vector machine classifier for the sake of identifying the best features that have the ability to optimize the detection of DNS tunneling. To evaluate the proposed method, a benchmark dataset of DNS tunneling has been used. Results showed that the proposed method has outperformed the conventional SVM by achieving 0.946 of F-measure.

Corresponding Author:

Fuqdan A. Al-Ibraheemi
College of Dentistry
University of Al-Ameed
Karbala PO Box 198, Iraq
Email: fuqdan@alameed.edu.iq

1. INTRODUCTION

Nowadays, the need for web to perform wide range of transactions is getting imperative for large organizations and end-users. This can be represented by the search, explore and reach operations conducted to access valuable information related to medical, financial and education purposes. Particularly, various businesses recently are considerably depending on the internet for conducting their daily-basis transactions. With such a dependency of internet, wide range of challenges would be posed. A significant aspect of such challenges is the information security where confidential data belong to critical parties such as medical and military would be exposed. Security threats take different forms, but one of these form is taking the advantage of domain name system (DNS) protocol for passing dangerous and malicious procedures, this attack attempt is known as DNS tunneling [1]. DNS is characterized by its simplicity where it intends to offer a straightforward way for accessing particular server through the domain name instead of the IP address [2-5]. Because of its simplicity, attackers attempt to use it for creating a tunnel to execute malicious scripts that intended to capture confidential information, gaining a super access, or attempting to harm the server [6].

Regardless, DNS is still an imperative component within the world wide web for conducting necessary operations such as the surfing and communications [7]. Its key success is represented by providing flexible way for reaching out a specific server through a straightforward name instead of the complicated form of IP address [8-12]. For instance, rather than typing a long and complicated combinations of IP addresses such as ‘120.90.10.1’, it is more convenient to type the ‘ServerDomainName.com’. However, such simplicity comes with a cost where the DNS is exposed to wide range of threats through tunnels. Therefore, powerful servers of well known organizations are vulnerable to such threats through the DNS tunneling [13]. This is due to the major consideration of other network security aspects which might yield greater threats compared to the DNS tunneling.

Wide range of methods were depicted in the literature for the identification of DNS tunneling [14]. One the traditional techniques is the Firewall where it has the ability to reject specific DNS queries based on some instructions. These instructions are simply comparing the IP address of specific DNS query in order to match it with preordained list of allowed IP addresses which might facilitate avoiding unknown DNS that yields harmful contents. Additionally, any intrusion detection system would also have the capability to detect DNS tunneling through the investigation of requests. Furthermore, analyzing the network traffic is another way of identifying DNS tunneling where extra features are being considered along with the domain name. Lastly, the passive replication technique has also the ability to avoid DNS tunneling through replicating each DNS query for denying the further use of them.

Yet, all the above-mentioned techniques are still having various drawbacks. This is because they are able to recognize few types of DNS tunneling. In addition, it is obvious that additional information is needed to detect a sophisticated DNS tunneling. Apparently, distinguishing the regular DNS queries from the DNS tunneling is a challenging task.

According to the insufficient performance of the traditional methods in terms of detecting DNS tunneling, recent studies have recommended the use of machine learning techniques in terms of detecting DNS tunneling [15-17]. Machine learning techniques have the ability to train and learn from previous experiences of DNS tunneling. Such learning paradigm has the ability to make the machine able to detect new types of DNS tunneling. Yet, there are some challenging issues that are facing the machine learning techniques in the process of identifying DNS tunneling.

One of the challenging tasks behind using the machine learning (ML) techniques for the sake of DNS tunneling detection is the high dimensionality of the features that could be used to differentiate the DNS queries from the tunneling ones. Basically, features play an essential role in the machine learning in which the significant features lead improve the performance of the classification and vice versa [18]. Selecting the most appropriate features was one of the major concerns within the community of machine learning researchers.

The features that have been exploited to identify the tunneling are categorized in two main classes payload analysis or traffic analysis [17]. Payload analysis can be used to detect DNS tunneling by analyzing the request and its features including domain length, number of bytes and content. Such features could be exploited to generate rules that intended to prevent DNS tunneling. Whereas, traffic analysis can be used to detect the DNS tunneling by examining the traffic and its features including volume of DNS traffic, number of hostnames per domain, geographic location and domain history. Apparently, there is a wide range of features can be exploited for the identification of tunneling through ML. Therefore, this paper intends to propose a feature selection approach based on genetic algorithm for DNS tunneling detection. The proposed feature selection approach will be combined with a support vector machine classifier.

2. RELATED WORK

The newly researches in DNS tunneling focuses on ML for the identification process. This is because ML would have the ability to learn from previous cases and generate statistical rules for future attempts. As an example of such studies is the one proposed by Dusi *et al.* [19] in which a system was designed to learn from previously known DNS tunneling attempts. The system was analyzing significant characteristics such as network traffic. Based on such features, the system will generate a set of rules to predict the occurrence of DNS tunneling.

In the same regard, Dusi *et al.* [14] presented an ML method aimed at detecting the tunnels within the application layer. Such method was intended to take the advantage of significant features to differentiate the normal and suspicious patterns. The features consisted of network traffic characteristics including connection duration and sending and receiving sizes. Lastly, an algorithm of decision tree (DT) has been trained on such features to detect the potential DNS tunneling. Results of classification demonstrated an outperformance over the conventional methods.

Beside the DT, Allard *et al*. [20] presented an algorithm of random forrest (RF) for preventing DNS tunneling. In a similar way to the previous study, this study has also taking the advantage of network traffic flow features. Lastly, the proposed classifiers have been trained on the feature space of flow characteristics and tested on a set of DNS query [20].

Aiello *et al*. [21] is another study where SVM algorithm has been used for the identification of tunneling. The authors have utilized a set of DNS queries and answers in order to train their model [21]. During the model building, queries' contents have been examined in terms of normal and suspicious patterns. Additionally, Aiello *et al*. [16] proposed a statistical fingerprint for the identification of DNS tunneling based on machine learning method. The proposed method has taken the advantage of connection features such as duration and size of packets.

Using RF algorithm, Buczak *et al*. [22] presented a model for preventing DNS tunneling. The proposed model has utilized the duration and size of the connections in order to initiate the feature space. Lastly, the proposed RF has been used to predict the occurrence of the DNS tunneling.

To examine a new form of features, Aiello *et al*. [23] utilized two feature extraction techniques including principle component analysis and mutual information. Such techniques aim at finding correlations and averaging the statistical features of network flow. Lastly, using a K-nearest neighbor algorithm, the authors have successfully managed to train a model for predicting the DNS tunneling.

Homem *et al*. [24] examined the capability of maximum entropy algorithm in terms of identifying DNS tunneling. The algorithm has been trained on traditional features such as connection duration and size. Lastly, a prediction model has been built to anticipate the occurrence of DNS tunneling.

Finally, Van Thuan Do *et al*. [15] have extended the feature space for predicting DNS tunneling within mobile networks. The authors have utilized duration of connection, length of DNS query and destination. Lastly, an SVM algorithm has been trained on such features. From the literature, that there are a wide range of features that have been addressed for the task of detecting DNS tunneling. In this manner, there is a vital demand to determine the most appropriate feature set. This is due to the imperative need of identify the best features.

3. PROPOSED METHOD

In order to identify the best feature set for detecting DNS tunnelling, this paper proposes a hybrid method of genetic algorithm (GA) feature selection approach with the support vector machine (SVM) classifier. To describe the application of the proposed solution, Figure 1 shows the workflow of the implementation.

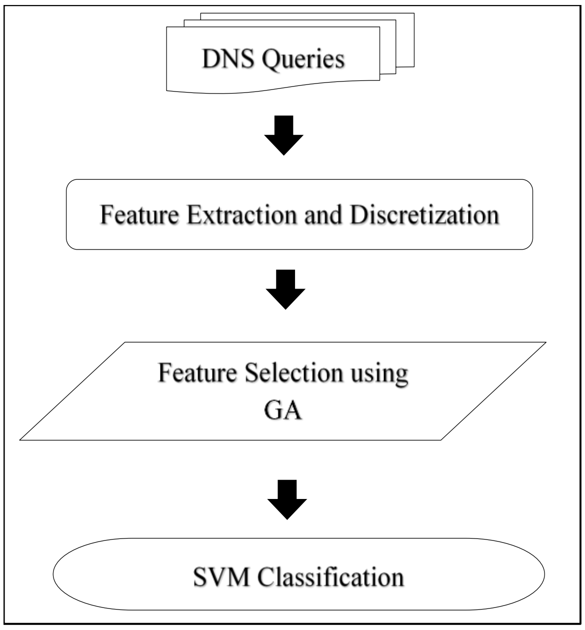

DNS Queries
Feature Extraction and Discretization
Feature Selection using GA
SVM Classification

Figure 1. Workflow of the proposed method

As shown in Figure 1, the implementation stage begins with a dataset of DNS queries that consists of legitimate and tunnelling queries. In order to establish the feature space, a feature extraction task will take a place in order to represent two types of features including payload and traffic features. Consequentially, GA will be used to search for the best solutions or in other word identifying the best features. Then, SVM will accommodate the classification task in which the queries will be divided into legitimate and tunnelling queries. However, the latter phases will be tackled in the next sections independently.

3.1. DNS tunneling dataset

In order to create the DNS tunnelled traffic dataset, [24] have simulated the use of different protocols regarding to their own DNS tunnel. The authors have developed a script in python programming language in order to generate and collect packet captures of network traffic by simulating the normal use of another two network protocols running over DNS tunnels including HTTPS and POP3. For both HTTP and HTTPS, the data was generated by visiting five random website. FTP was simulated through the directory traversal and downloading five random files from an FTP server. Finally, POP3 protocol was simulated by requesting and downloading five random emails from a mail server in which some of these email contains plain text and the other contains randomly generated files as attachments. Table 1 depicts the details of the DNS tunneling dataset.

Table 1. Dataset details

Network Protocol	Number of Samples	Original Size	Reduced Size
HTTP	52	636.7 MB	179.9 MB
HTTPS	53	686.3	181.4 MB
FTP	53	260.2 MB	49.8 MB
POP3	53	131.6 MB	28.2 MB
Total	211	1714.8 MB	436.3 MB

3.2. Feature extraction

In fact, features play an essential role in the context of supervised machine learning in which the strong features would significantly improved the performance of the classification, vice versa; the weak features would negatively affect the performance of the classification. In this manner, in order to generate features for the network traffic, [24] have analysed different features related to both normal and tunnel traffic. In fact, numerous features can be extracted by observing both normal and tunnel traffic such as byte frequencies, information entropy and packet lengths. However, one of the significant feature is the information entropy due to its direct correlation with the actual data bytes composing packets.

Investigating the network flow features would include various levels such as IP-level, transport-level or application-level. The key difference among such levels lies between the client request and server answer. For instance, some protocols would have similar or even identical content regarding the request's content. Yet, the key distinguish between them lies on the server response. Therefore, concentraing on the contenct of response is considered much appropriate to examine the variances. In particular, some techniques such as information entropy would be a great help in terms of identifying the variation within a message. One way to compute the entropy is by estimating the bytes within a packet [24]. Hence, the probability of certain occurrences of byte $p(x_i)$ multiplied by its logarithm would contribute toward identifying the entropy as in the following formula:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \times log\, p(x_i) \quad (1)$$

The primal benefit of acquiring entropy lies behind gaining an insight regarding the distribution produced within every protocol flow. Hence, the distribution's trend is acquired through such entropy analysis. Along with the information entropy, there are other features that were provided for each connection such as DNS request length, IP packet sender length, IP packet response length, encoded DNS query name length, request application layer entropy, IP packet entropy and query name entropy.

3.3. Discretization

In mathematics, discretization task aims to transfer continuous functions, models and equations into discrete values. For machine learning, the values are either nominal or numeric. Nominal values are like predefined ranges such as 'low, medium, high' or 'black, white'. In this manner, the nominal values tend to be discrete. For the numeric values, there are some case where it can be discrete such as the values from 0 to 9 which can be ranged in ten discrete values of {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. However, sometimes the numeric values can be formed as continuous rather than discrete. In this vein, it is difficult to range or limit the continuous values due to the fluctuation of numbers. Therefore, it is necessary to accommodate a discretization task in order to put these values within a range. Table 2 shows an example of discretization of continuous values.

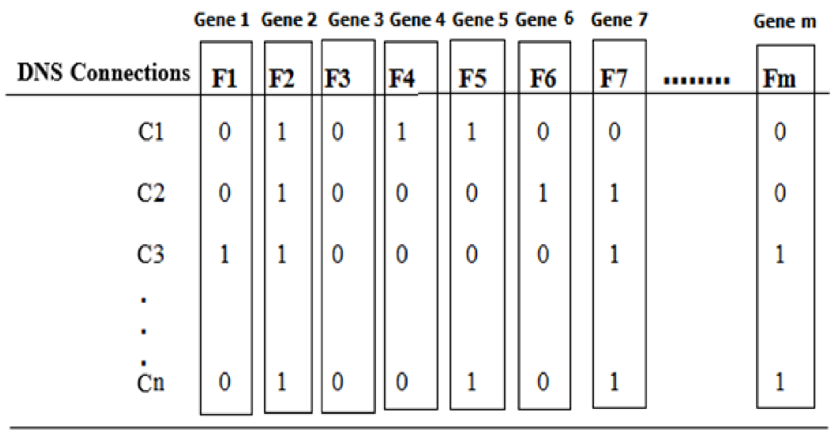
Table 2. Example of discretization

Instance	Continuous value	Convert rules
I1	5.30498	IF value < 5.30500 ➔ A
I2	5.30500	IF value > 5.30500 & value < 5.30510 ➔ B
I3	5.30515	IF value > 5.30510 & < 5.30520 ➔ C
I4	5.30529	IF value > 5.30520 & < 5.30530 ➔ D
I5	5.30535	IF value > 5.30530 ➔ E

3.4. Feature selection using genetic algorithm

In this section, the features of the DNS connections will be examined in order to identify the most appropriate features that have the ability to distinguish the occurrence of tunnelling. For this purpose, a meta-heuristic approach of genetic algorithm has been used. Genetic algorithm (GA) is one of the local search techniques that simulate the biology of natural selection [25, 26]. Theoretically, it works by generating a population where the possible solutions can be examined. Consequentially, a combination process is performed by combining the best population based on a fitness function.

In fact, genetic algorithm starts with producing an initial population where the possible solutions can be represented. This population consists of different genes (i.e. features) that composed of chromosomes. Each chromosome is represented by a binary value (i.e. 0 or 1). Figure 2 depicts the representation of these genes (features) where C refers to a single DNS connection, F refers to a feature and m refers to the number of features.



	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7		Gene m
DNS Connections	F1	F2	F3	F4	F5	F6	F7	……..	Fm
C1	0	1	0	1	1	0	0		0
C2	0	1	0	0	0	1	1		0
C3	1	1	0	0	0	0	1		1
.									
.									
.									
Cn	0	1	0	0	1	0	1		1

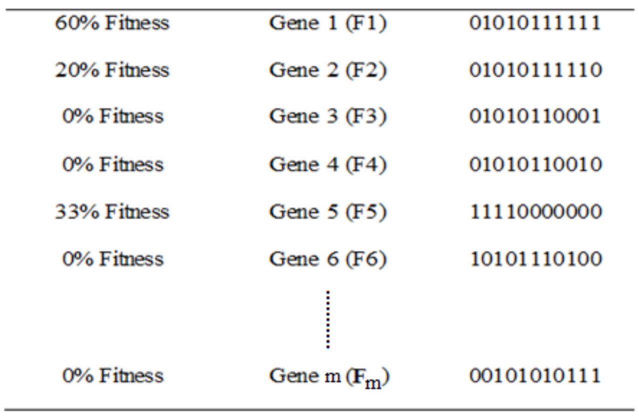
Figure 2. Initial population of genes

Once the producing of the initial population is being done, each gene have to be assessed in terms of feasibility. This assessment is conducted based on fitness function which identifies the desired effectiveness using a value. Figure 3 depicts the evaluation of genes with random of values of fitness.

As shown in Figure 3, there are genes that have low value of fitness such as Gene 3, Gene 4, Gene 6 and Gene 7 which have 0% of fitness. Hence, genetic algorithm will get rid of these genes regarding to their infeasibility. At the same time, Gene 1, Gene 2 and Gene 5 will be selected as the best genes regarding their effectiveness in terms of fitness. Note that, there are multiple strategies to select the best genes such as roulette wheel, rank selection, steady-state selection. However, in this study the rank selection strategy has been utilized.

Once the best genes are being selected, a re-production process is performed which aims to generate the next population. This can be conducted by manipulating the chromosomes of the current genes. For this purpose, there are multiple methods can be used to re-produce the genes such as crossover, mutation and elitism. In this study, crossover method has been utilized to re-produce the next generation.

Crossover is an exchange procedure that aims to swap the chromosomes between two or more genes. This process aims to combine the strongest chromosomes from two genes and formulate it in a new gene for next population. Figure 4 depicts the mechanism of crossover. The specification of applying the genetic algorithm can be seen in Table 3.



60% Fitness	Gene 1 (F1)	01010111111
20% Fitness	Gene 2 (F2)	01010111110
0% Fitness	Gene 3 (F3)	01010110001
0% Fitness	Gene 4 (F4)	01010110010
33% Fitness	Gene 5 (F5)	11110000000
0% Fitness	Gene 6 (F6)	10101110100
	⋮	
0% Fitness	Gene m (F_m)	00101010111

Figure 3. Evaluating genes based on the fitness function

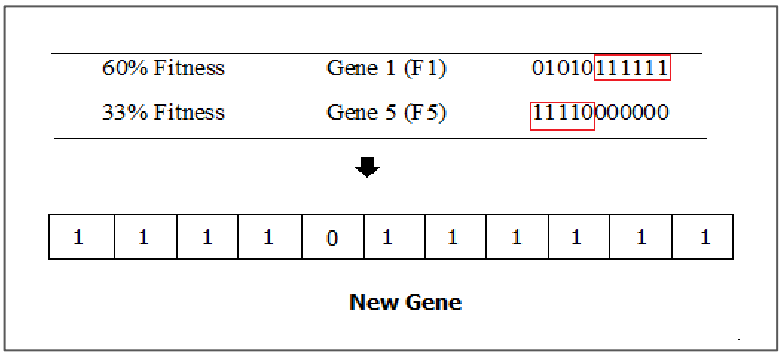


Figure 4. Crossover exchange

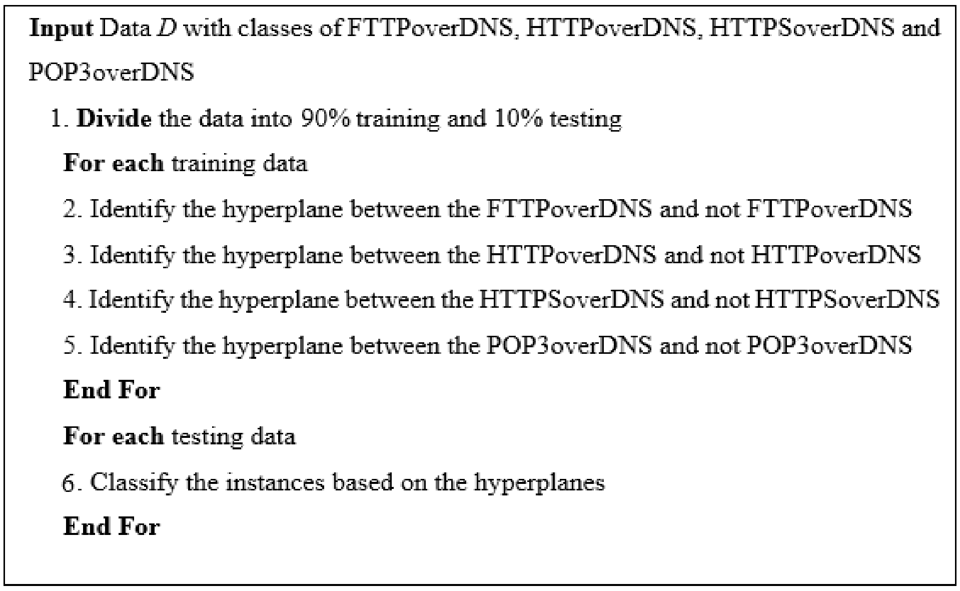
Table 3. Details of carrying out the genetic algorithm

Parameter	Description
Number of iterations or generations	50
Re-production mechanism	Crossover
Condition for termination	No significant changes in next generations

3.5. Classification using SVM

In this section, the classification of the DNS connection is being performed in which each connection would be classified into its actual class label. Such classification process has been conducted using the support vector machine. This classifier is a non-probabilistic and binary classifier in which the data is partitioned into two groups (0 or 1). The work mechanism of this classifier lies on assigning a hyperplane which is a margin that partition the data into the two groups [27]. Note that, SVM will classify the connection instances based on the selected features by the genetic algorithm in which the training portion was 80% and the testing portion was 20%. Figure 5 depicts the algorithm of SVM classifier.

As shown in Figure 5 basically, will identify a margin that partition the data with the first class (i.e. FTTPoverDNS) and the remaining data (shown in step 2). After that, identifying a margin that partition the data with the second class (i.e. HTTPoverDNS) and the remaining data (shown in step 3). Consequentially, identifying a margin that partition the data with the third class (i.e. HTTPSoverDNS) and the remaining data (shown in step 4). After that, identifying a margin that partition the data with the fourth class (i.e. POP3overDNS) and the remaining data (shown in step 5).



```
Input Data D with classes of FTTPoverDNS, HTTPoverDNS, HTTPSoverDNS and
POP3overDNS
  1. Divide the data into 90% training and 10% testing
   For each training data
   2. Identify the hyperplane between the FTTPoverDNS and not FTTPoverDNS
   3. Identify the hyperplane between the HTTPoverDNS and not HTTPoverDNS
   4. Identify the hyperplane between the HTTPSoverDNS and not HTTPSoverDNS
   5. Identify the hyperplane between the POP3overDNS and not POP3overDNS
   End For
   For each testing data
   6. Classify the instances based on the hyperplanes
   End For
```

Figure 5. SVM algorithm

4. RESULTS AND DISCUSSIONS

Since this study is concentrating on the supervised machine learning thus, the evaluation method that has been used for assessing the classification performance will be used. Several research studies have addressed the evaluation of classification using the common information retrieval metrics including precision, recall and F-measure [24]. To compute such metrics, the contingency table will be used as shown in Table 4.

Table 4. Contingency table

		Predicted DNS connection	
	Predicted / Actual	Normal	Tunnel
Actual DNS connection	Normal	*True positive (TP)*	*False positive (FP)*
	Tunnel	*False negative (FN)*	*True negative (TN)*

False negative (FN): is the number of actual tunneling connections that have been predicted as normal.
False positive (FP): is the number of actual normal connection that have been predicted as tunneling.
True negative (TN): is the number of correctly un-predicted connections.
True positive (TP): is the number of correctly predicted connections.

In this manner, the precision, recall and F-measure can be computed based on the following equations.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+TN} \tag{3}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Precision is the ratio between the number of correctly classified connections over the total number of connections. While Recall is the ratio between the number of correctly classified tunneling connections and the total number of tunneling connections. Finally, F-measure is considered to be the total accuracy. Basically, the evaluation will be held based on the hybrid of SVM and GA compared to the conventional SVM. Table 5 shows such comparison.

Table 5. Results of the proposed method compared to the conventional SVM

Class	F-measure	
	SVM	SVM & GA
FTP	0.82496	0.90989
HTTP	0.78981	0.98989
HTTPS	0.83742	0.91431
POP3	1.0	0.97
Average	0.86305	0.94602

As shown in Table 5, the use of genetic algorithm with the support vector machine classification has significantly improved the F-measure values for all most of the class labels. First, the FTP has been enhanced from 82% of F-measure into 90%. Second, HTTP class label has been improved from 78% into 98% of F-measure; this is the dramatic enhancement that occurred upon a class label compared to the others. For the HTTPS class label the enhancement was slight in which the F-measure has been increased from 83% to 91%. However, for the POP3 class label, the result of F-measure has been decreased from 100% into 97%. Although the use of GA has not enhanced the classification of POP3 however, the average performance was significantly improved from 86% to 94% of F-measure. Generally, the use of genetic algorithm has significantly improved the classification performance of DNS connections compared to the traditional classification without applying GA. This can imply the usefulness of GA where the capabilities of identifying the most appropriate feature set has led to enhance the classification accuracy. Figure 6 compares the performances of SVM and SVM with GA.

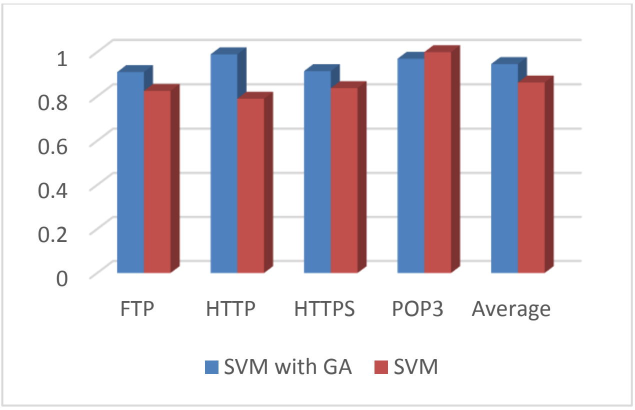


Figure 6. Performances of SVM and SVM with GA

On the other hand, it is necessary to compare the results of the proposed solution with the state of the art. Our baseline study was the one that proposed by Homem and Papapetrou [28] who applied the SVM classification on the same benchmark dataset that has been used in this study and achieve an F-measure of 91%, it is obvious that our proposed method has outperformed the results of the state of the art by achieving 94% of F-measure. This can demonstrate the hypothesis of our study in which the use of the genetic algorithm in order to identify the best features will lead to enhance the effectiveness of the classification.

5. CONCLUSION

This paper has proposed a hybrid method of genetic algorithm and support vector machine for the process of detecting DNS tunneling. The proposed GA has demonstrated substantial performance in terms of selecting the best features which let to improve the classification accuracy. Addressing different feature selection approaches in future researches is a key challenging issue in which ant colony, particle swarm optimization and simulated annealing are competitive feature selection methods compared to GA.

REFERENCES

[1] T. van Leijenhorst, et al., "On the viability and performance of DNS tunneling," University of Wollongong Research Online, 2008.

[2] A. Sani and M. Setiawan, "DNS tunneling Detection Using Elasticsearch," in *IOP Conference Series: Materials Science and Engineering*, vol. 722, 2020, pp. 1-9.

[3] N. Ishikura, et al., "Cache-Property-Aware Features for DNS Tunneling Detection," in *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2020, pp. 216-220.

[4] H. Bai, et al., "Refined identification of hybrid traffic in DNS tunnels based on regression analysis," *ETRI Journal*, pp. 1-13, 2020.

[5] H. Ichise, et al., "NS record History Based Abnormal DNS traffic Detection Considering Adaptive Botnet Communication Blocking," *Journal of Information Processing*, vol. 28, pp. 112-122, 2020.

[6] K. Born and D. Gustafson, "Ngviz: detecting dns tunnels through n-gram visualization and quantitative analysis," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, p. 47, 2010.

[7] K. Born and D. Gustafson, "Detecting dns tunnels using character frequency analysis," *arXiv preprint arXiv: 1004.4358*, 2010.

[8] K. Bumanglag and H. Kettani, "On the Impact of DNS Over HTTPS Paradigm on Cyber Systems," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, 2020, pp. 494-499.

[9] G. Yan, et al., "Discovering Suspicious APT Behaviors by Analyzing DNS Activities," *Sensors*, vol. 20, no. 3, p. 731, 2020.

[10] F. Palau, et al., "Detecting DNS Threats: A Deep Learning Model to Rule Them All," in *XX Simposio Argentino de Inteligencia Artificial (ASAI 2019)-JAIIO 48*, Salta, 2019.

[11] A. Almusawi and H. Amintoosi, "DNS Tunneling Detection Method Based on Multilabel Support Vector Machine," *Security and Communication Networks (Hindawi)*, vol. 2018, p. 9, 2018.

[12] M. Sammour, et al., "DNS Tunneling: a Review on Features," *International Journal of Engineering and Technology*, vol. 7, no. 20, pp. 1-5, 2018.

[13] R. Rasmussen, "Do you know what your dns resolver is doing right now," *Security Week*, 2012. [Online]. Available: http://www.securityweek.com/do-you-know-what-your-dnsresolver-doing-right-now.

[14] M. Dusi, et al., "Tunnel hunter: Detecting application-layer tunnels with statistical fingerprinting," *Computer Networks*, vol. 53, no. 1, pp. 81-97, 2009.

[15] P. E. Van Thuan Do, et al., "Detection of DNS Tunneling in Mobile Networks Using Machine Learning," *Information Science and Applications 2017 (ICISA 2017)*, vol. 424, 2017, p. 221.

[16] M. Aiello, et al., "DNS tunneling detection through statistical fingerprints of protocol messages and machine learning," *International Journal of Communication Systems*, vol. 28, no. 14, pp. 1987-2002, 2015.

[17] G. Farnham and A. Atlasis, "Detecting DNS tunneling," *InfoSec Reading Room*, 2013.

[18] S. B. Kotsiantis, et al., "Supervised machine learning: A review of classification techniques," *Informatica*, pp. 249-268, 2007.

[19] M. Dusi, et al., "Detection of encrypted tunnels across network boundaries," in *IEEE International Conference on Communications (ICC'08)*, 2008, pp. 1738-1744.

[20] F. Allard, et al., "Tunneling activities detection using machine learning techniques," DTIC Document, 2010.

[21] M. Aiello, et al., "Basic classifiers for DNS tunneling detection," in *2013 IEEE Symposium on Computers and Communications (ISCC)*, 2013, pp. 000880-000885.

[22] A. L. Buczak, et al., "Detection of Tunnels in PCAP Data by Random Forests," in *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, 2016, p. 16.

[23] M. Aiello, et al., "Profiling DNS tunneling attacks with PCA and mutual information," *Logic Journal of IGPL*, vol. 24, no. 6, pp. 957-970, 2016.

[24] I. Homem, et al., "Entropy-based Prediction of Network Protocols in the Forensic Analysis of DNS Tunnels," *arXiv:1709.06363*, 2016.

[25] Y. Song, et al., "An improved genetic algorithm for numerical function optimization," *Applied Intelligence*, vol. 49, no. 5, pp. 1880-1902, 2019.

[26] R. Vijayanand, et al., "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers & Security*, vol. 77, pp. 304-314, 2018.

[27] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[28] I. Homem and P. Papapetrou, "Harnessing Predictive Models for Assisting Network Forensic Investigations of DNS Tunnels," *ADFSL Annual Conference on Digital Forensics, Security and Law*, 2017, pp. 1-11.