Computer Networks 160 (2019) 118–129

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A multi-dimensional machine learning approach to predict advanced malware

Şerif Bahtiyar*, Mehmet Barış Yaman[1], Can Yılmaz Altınığne[1]

Department of Computer Engineering, Istanbul Technical University Maslak, Istanbul, 34469, Turkey

ARTICLE INFO

Article history:
Received 18 October 2018
Revised 23 February 2019
Accepted 13 June 2019
Available online 13 June 2019

Keywords:
Advanced malware
Machine learning
API Call
Prediction
Classification

ABSTRACT

The growth of cyber-attacks that are carried out with malware have become more sophisticated on almost all networks. Furthermore, attacks with advanced malware have the greatest complexity which makes them very hard to detect. Advanced malware is able to obfuscate much of their traces through many mechanisms, such as metamorphic engines. Therefore, predictions and detections of such malware have become significant challenge for malware analyses mechanisms. In this paper, we propose a multi-dimensional machine learning approach to predict Stuxnet like malware from a dataset that consists of malware samples by using five distinguishing features of advanced malware. We define the features by analyzing advanced malware samples in the wild. Our approach uses regression models to predict advanced malware. We create a malware dataset from existing datasets that contain real samples for experimental purposes. Analyses results show that there are high correlations among some features of advanced malware. These provide better predictions scores, such as $R^2 = 0.8203$ score for Stuxnet closeness feature. Experimental analyses show that our approach is able to predict Stuxnet like advanced malware if prediction features defined.

1. Introduction

The amount and variety of attacks on computing systems including all types of networks increase in an enormous speed. This trend is driven by a rich volume of different malware. The richness has a huge impact on the cost of computing systems. Actually, the cost depends on the success of attacks. Advanced malware has become an effective tool to accomplish such attacks.

Advanced malware is a complex malicious software which has very effective properties. The main purpose of such malware is to accomplish targeted attacks with high success ratio. Specifically, critical systems are main targets of advanced malware. This type of malware uses different attack vectors to accomplish its goal and it has exceptionally complex structure [1]. Moreover, advanced malware may use conventional malware to increase the success ratio, such as using ransomware [2]. Therefore, many systems and networks have suffered from advanced malware considerably. For instance, financial systems and critical networks are the targets of such malware [3,4].

Recently, malware is used in many complex targeted attacks. Existing anti-malware systems and intrusion detection systems are able to detect some traces of attacks if they are carried out with conventional malware. In this paper, we distinguish malware in two categories, namely conventional and advanced malware as in [5]. Conventional malware is malicious software that are already categorized in literature, such as virus, worm, and etc. [6]. Moreover, this type of malware is almost always detectable with adequate anti-malware systems [7]. On the other hand, advanced malware has been undetectable until the attack is completed [5].

The grand challenge is to predict and detect advanced malware before it completes its tasks. New detection and prediction mechanisms are needed for these purposes since existing anti-malware systems have not detected such malware yet. In this paper, machine learning algorithms are used to extract information for predicting and detecting advanced malware based on features of conventional and advanced malware instances seen in the wild. Stuxnet is the first advanced malware known in the wild. In this paper, we only consider Stuxnet like malware as advanced ones. Our main contributions are as follows.

- We analyze and find distinguishing properties of Stuxnet like advanced malware by using malware samples seen in the wild. We expect that the properties may be used with machine learning algorithms to identify the type of malware. We propose an approach to predict advanced malware by using these properties.

* Corresponding author.
E-mail address: bahtiyars@itu.edu.tr (Ş. Bahtiyar).
[1] Authors equally contributed to this work.

https://doi.org/10.1016/j.comnet.2019.06.015
1389-1286/

Table 1
Comparison of malware categories.

Properties	Advanced malware	Combo malware	Conventional malware
Stealth	Use of stolen signature and others	Use methods of conventional malware	Depends on the type of malware
Creation	Codes from existing and unknown malware	Borrowing codes from existing malware	Generating code from known malware
Size	Generally bigger size	Sum of components size	Smaller size
Propagation	Many methods such as fragmentation	Conventional malware components	Depends on the type of malware

- Advanced malware uses conventional malware during an attack. We extract correlations between features of conventional malware and advanced malware. Additionally, we use the correlations to predict the type of malware namely, conventional or advanced. We believe that the predictions will be used to counter against attacks with advanced malware.
- We analyze a dataset with our approach for predicting the type of malware. The dataset contains only malware samples that are represented with API calls. Our analyses results on the dataset show that machine learning algorithms will be used to predict the type of malware by investigating the correlations among features.

To the best of our knowledge, this is the first work to predict Stuxnet like advanced malware by using only malware dataset, which does not contain benign samples.

The rest of the paper is organized as follows. Section 2 contains the evaluation of malware and the state of art about machine learning based malware detection and prediction. Next section is devoted to our model for the prediction of advanced malware. Section 4 is about analysis of malware with machine learning algorithms. We conclude the paper in the last section.

2. Malware and machine learning

In this section, we briefly explain malware types and machine learning approaches to detect and predict malware. We categorize malware according to its properties for extracting distinguishing features of advanced malware as in [5], which may be used with machine learning mechanisms for prediction and detection purposes. Table 1 contains a comparison of malware categories. Additionally, we present the state of art about machine learning based malware detection and prediction.

2.1. Malware

Malware is a malicious software used to deliberately harm computer systems, harvest critical data and system resources, manipulate network transactions and access private information of individuals [8]. Worm, virus, Trojan horse, spyware, botnet, and rootkit are instances of conventional malware. We define conventional malware as malicious software that is detectable by some anti-malware systems and all properties of it are well defined.

Malware has become more complex than ever that sometimes makes the categorization difficult. For instance, there are malware that may fall into different categories. For instance, Stuxnet is categorized as either worm or advanced malware. Therefore, we intentionally explain conventional malware types and advanced malware.

Viruses are malicious software that can replicate themselves whenever they are active. Virus needs a host to survive so it is primitive malicious software. On the other hand, worm like malware is a stand-alone malicious software [9]. Each malware has a goal, but virus may have a simple goal. For example, virus infected software may give rise errors to the system [6].

There are malware types that provides stealthy property for other malware types, such as Trojan horse. This malware is used to help infection mechanisms of other malware or it is used to steal information from infected host. Specifically, Trojan horse provides remote access to the infected systems. Unlike worm and virus, Trojan horse does not replicate itself [9]. This type of malware is sometimes considered as an espionage malware.

Key-loggers monitor information from the targeted system by recording keystrokes on the infected machine. There many types of key-loggers. Spyware affects the system or machine to monitor a wide range of critical information [10]. Backdoor or trapdoor may settle as a part of the system. Moreover, this type of malware may gain access to the system by providing authentication controls to the owner of malware.

One of the most effective malware to provide stealthy property is rootkit. It inserts a set of software codes to the targeted system for gaining administrators privileges for a remote control purpose while hiding its existence [11].

Similar to rootkit, a botnet has remote control facility. Botnets are remotely controlled computer network systems. These networks may be used for different purposes, such as sending spam e-mails or carrying out denial of service attacks [12].

Combo malware is a combination of many conventional malware. For instance, combo malware may consist of virus and worm. This type of malware is created by borrowing code from existing conventional malware [13]. For example, Lion and Bugbear.B malware may be categorized as combo malware. Lion malware is composed of Linux worm and rootkit. On the other hand, Bugbear.B is a combination of worm, virus, and backdoor [13]. Thus, their payload, size, and propagation depend on the components accordingly. All these malware types are detectable with some anti-malware mechanisms that are already running in the wild.

Advanced malware is sophisticated malicious software that has exceptionally different structure than conventional malware. The major properties of advanced malware are complexity, goal orientation, modular, stealth, being written in multi-languages, use of cryptography, and use of multiple vulnerabilities [5]. Advanced malware has dynamic nature therefore its components, infection mechanisms, and payload properties may change over time. Moreover, initial instances of advanced malware have greater size than conventional malware. These make the detection of advanced malware almost always impossible with conventional malware detection tools. On the other hand, advanced malware and conventional malware has some common properties that may be used to detect the presence of advanced malware on a specific host.

Stuxnet, Duqu, Flame, and Red October are the most common examples of advanced malware seen in the wild [14]. They share some common properties but they also have some differences. For example, Stuxnet is the one which infects removable drivers, local area networks, programmable logic controllers(PLC). It exploits vulnerabilities of systems that do not provide secure message verification and source authentication [15]. Additionally, Stuxnet has rootkit functionality, self-replication property over the network and it infects programs and uses encryption methods. The main purpose of Stuxnet is sabotage [14].

Duqu was found in September 2011 by CrySys. It has Command and Control servers. Moreover, Duqu has an auto destruction component that is based on time triggering mechanism. Duqu measures time taken after the infection. In some resources, it is

considered as stealthy spyware [16]. Interestingly, it has only manual replication property. Additionally, Duqu uses AES algorithm for encryption operations. The main goal of Duqu is information gathering [14]. In other words, its goal is espionage.

Flame was discovered in May 2012 and it is different than the other advanced malware in terms of the size, which is approximately 20 megabytes [17]. Like Duqu, Flame replicates itself manually. It uses encryption mechanisms for different purposes, such as to provide stealthy property. The main design purpose of Flame is information gathering [14].

Red October was discovered in October 2012 in the wild and it is considered as active since May 2007. Initially, Red October infected Microsoft Office programs and Java. Similar to Flame, this malware is replicated manually. It has also keylogging module and encryption property. It is used for espionage purpose [14]. These properties give rise to thought that Red October and Flame were designed by same developers. Additionally, Red October is considered as a cyber weapon since it has additional properties than conventional malware.

2.2. Malware analysis with machine learning

The traditional approaches of security mechanisms are unable to prevent advanced cyber-attacks because of the complex nature of new generation malware [18]. One of the challenges to prevent such attacks is a lack of intelligences about advanced malware. On the other hand, there is an awareness for the need of Technical Threat Intelligence (TTI) about advanced malware and corresponding attacks [19]. Thus, predicting and detecting malware has become a significant challenge to prevent advanced attacks.

Cyber-attacks with malware have taken attentions of nation states and some organizations. They have used the attacks for either sabotage or espionage purposes. These kinds of attacks are sometimes referred as Advanced Persistent Threat (APT) [20]. Moreover, there are classification or detection solutions to cope with such malware that are based on machine learning [21] and data mining [22]. However, each of them considers only specific features or environments, such as self-organizing feature [23] and malicious URL [24].

The vast number of different computing platforms and their interconnections make malware detections with machine learning more challenging than ever. There are many machine learning based researches to classify and detect malware [23,25]. The researches take into account only some features of malware [26]. They do not consider the detection of all the types with a common approach or machine learning algorithm. For example, some researches considers only Android platforms [27] whereas some others take into account Windows platform [25].

There are researches that use multiple features to detect malware with specific machine learning approaches. For instance, a deep learning based method to detect Android malware uses many features, such as similarity based ones [28]. Another approach uses deep learning to classify malware features for the identification [29]. Layton et al. considers API calls that identifies and predict banking malware [30]. Moreover, machine learning approaches are designed to predict APTs by using a correlation framework [31]. To the best of our knowledge, there is no research that distinguish type of malware as either conventional or advanced based on a dataset that consists only of malware instances. The goal of our approach is to predict Stuxnet like advanced malware based on multiple features that are represented with API calls.

3. A model to predict advanced malware

In this paper, our goal is to predict advanced malware that is similar to Stuxnet. The prediction model is based on distinguishing properties of Stuxnet. We intentionally selected the properties that are related to specific conventional malware to be able to use existing malware datasets. After careful analysis of advanced malware, we define five features that are represented with API calls. Conventional malware arsenal, behavior instability, stealthy property, metamorphic engine, and closeness to Stuxnet are our features. We use them to distinguish advanced and conventional malware by extracting relationships among features.

We use regression algorithms to show correlations among features of advanced malware for predictions. Fig. 1 shows the high-level algorithm for the prediction. The features are also expected to be used for the detection purpose of advanced malware in anti-malware systems. Advanced malware may have both new features and features of conventional malware. Since we have a limited number of advanced malware known in the wild, we have enriched our features with conventional malware features to increase the prediction accuracy. For instance, conventional malware arsenal represents many activities of malware in the wild. Therefore, we have considered both distinguishing features of advanced malware and properties of conventional malware.

3.1. Machine learning models

We use regression algorithms to extract relationships among features of malware. In this paper, linear, polynomial, and random forest regression models are used to determine the relationships. Moreover, we use relationships among malware features for prediction purposes. Here are brief explanations of the models.

- Linear regression model: This type is used to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable. Other one is a dependent variable. Relationships between two variables are modeled using a linear predictor function. In this model, unknown parameters are estimated by using data from which some features are extracted.
  The regression model considers a dataset that consists of explanatory variable namely, independent variable, x and a dependent variable y. The relationship between y and x depends on x and constant β, which is called the *intercept*.
  $$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + + \beta_n x_n + \epsilon_i \quad (1)$$
  In real life, there is always a disturbance between x_i and y_i that is called error variable. Most of the time error is represented with ϵ_i as in Eq. (1).
- Polynomial regression model: In this model, the relationship between independent variable x and dependent variable y is modeled as an nth degree polynomial in x. The regression fits a nonlinear relationship between x and the corresponding conditional mean of y. Formal representation of polynomial regression model is as follows.
  $$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n + \epsilon \quad (2)$$
  Similar to linear regression model, ϵ represents an unobserved random error.
- Random forest regression model: This model has a good behavior to handle more complex relationships among features of data. Random forest models capture non-linear interaction between the known features and the targeted one, which differs this model from linear regression models. This model is an additive model that makes predictions by combining decisions from base models. Formal representation of the model is as follows.
  $$g(x) = f_0(x) + f_1(x) + f_2(x) + \ldots \quad (3)$$
  In Eq. (3), $g(x)$ is the sum of simple base models $f_i(x)$. Each base classifier is a simple decision tree. In random forest models,

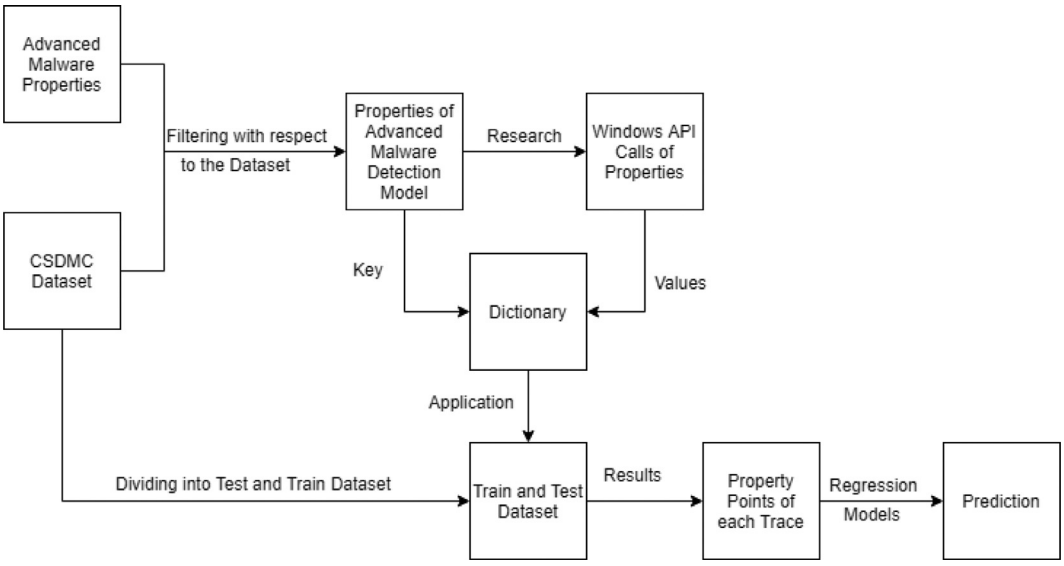


Fig. 1. The high level algorithm to predict advanced malware with machine learning.

all base models are constructed independently using a different subsample of data.

3.2. Prediction model

We use relationships among five features of malware for the prediction of malware type namely, conventional or advanced. Our prediction model is based on correlations among these five properties of advanced malware. We represent malware with set M that is related to features X_i as follow:

$$M = \{X_1, X_2, X_3, X_4, X_5\} \tag{4}$$

The first feature is conventional malware arsenal that represents many activities of traditional malware, such as screen capture, anti-debugging, downloader, DLL injection and dropper [32]. Advanced malware uses conventional malware for many different purposes, therefore, the arsenal is one of the five features. We use this feature to predict advanced malware because Stuxnet contains these kind of properties.

Behavior instability is a significant feature to distinguish advanced malware. In our model, we take into account read/write files, search file for an infection purpose, load a register, modify file attributes, get file information, distribute global/virtual memory, copy/delete files, and access to files as in [33,34]. We analyzed this feature to show its correlations with other features since all features of advanced malware are somehow dependent to each other. Thus, behavioral instability is our second feature in the proposed model.

Stealthy property is one of the key characteristics of advanced malware. It has seen from many instances of advanced malware in the wild. For instance, Stuxnet has lots of propagation mechanisms and anti-malware systems are unable to detect it because of stealthy property in propagation mechanisms. Advanced malware uses many hiding techniques, such as rootkit hiding mechanisms [35,36]. Most of the time, stealthy property increases the success ratio of attacks for advanced malware. In our model, we use stealthy property to distinguishing the type of malware. Therefore, this the third feature of our model.

Next feature is metamorphic or polymorphic engine. Having a metamorphic engine helps to hide the traces of advanced malware [37]. Therefore, this property is directly related to stealthy property. Malware uses this feature to make detections difficult by anti-malware systems. Therefore, extracting correlations among these engines are a key to design prediction and detection mechanisms for advanced malware.

In this work, we consider advanced malware that is similar to Stuxnet therefore our last feature is Stuxnet closeness. Actually, many instances of advanced malware share similar features with Stuxnet. Therefore, we observed that if malware is close to Stuxnet, this malware is also close to be advanced. For instance, Stuxnet has some rootkit functionality, XOR encryption, DLL PE excitability, and self-replication over the network. Moreover, it uses removable devices [14]. This feature is directly related to properties of advanced malware since Stuxnet is probably the first advanced malware known in the wild. Furthermore, there are evidences that recent advanced malware contains some modules of Stuxnet.

Advanced malware may be designed for multi platforms. The proposed model is for Windows platforms and the five properties are represented with Windows API calls. Fig. 1 shows the high-level algorithm to predict advanced malware. We compute the prediction of advanced malware with Eq. (7). Specifically, there are two metrics related to each malware in the computation, $S_i(X_i, P)$ and $D_i(X_i, P)$. Particularly, $S_i(X_i, P)$ is the ratio of number of API calls from class X_i related to malware P to total number of API calls in class X_i. In our model, malware P consists of API calls.

$$S_i(X_i, P) = \frac{\sum_{\forall j \in P \wedge j \in X_i} x_{i,j}}{\sum_{\forall k \in X_i} x_{i,k}} \tag{5}$$

In Eq. (7), $x_{i,j} = 1$ if corresponding API call exists, otherwise $x_{i,j} = 0$. $D_i(X_i, P)$ is the ratio of number of API calls from class X_i related to malware P to total number of API calls of P as shown in Eq. (6).

$$D_i(X_i, P) = \frac{\sum_{\forall j \in P \wedge j \in X_i} x_{i,j}}{\sum_{\forall k \in P} x_{i,k}} \tag{6}$$

$$A(X_i, P) = \prod_{\forall X_i, X_i \in M} S_i(X_i, P) D_i(X_i, P) \tag{7}$$

$A(X_i, P)$ is the closeness score for malware P and property X_i to Stuxnet like malware according to predicted feature. For instance, $A(X_3, P)$ represents malware P closeness to stealthy property. In these equations, all values are between zero and one. The value of one represents maximum closeness whereas the value of zero represents no correlation. Moreover, if $A(X_3, P) = 0.3$, we can say that stealthy property score of malware P is 0.3.

Our analysis of Stuxnet and related advanced malware show that this type of malware has high closeness score among the five features. Therefore, our prediction model is based on closeness scores between two or more than two features. Specifically, we compute a closeness score with the regression models related to a specific feature. If the score has a values above the threshold, we classify malware as advanced, otherwise as conventional. Since we do have limited amount of advanced malware instances known in the wild, we have not specified a certain threshold in this paper. Actually, the most significant challenge for the prediction of advanced malware is the find closeness scores therefore in this paper, we consider the computation of scores.

4. Analysis of malware features with machine learning

Malware analysis has never been done with an exact approach because of the everlasting evaluation of malware types and its capabilities. Moreover, there is little information about advanced malware that makes it impossible for an anti-malware system to predict and then detect such malware. On the other hand, there are prediction mechanisms whether a code is malicious or benign [38] in which case predicting the type of malware has become a significant challenge. Analyses of this paper are devoted to predict Stuxnet like advanced malware from a dataset that contain only malware samples. We consider Stuxnet like advanced malware for predictions which may leave traces like conventional malware since there is a lack of information for their exact detections.

We analyzed malware instances by extracting correlations among features of malware to predict potential advanced malware. Our analysis is based on five distinguishing features of Stuxnet and the goal of this analysis is to predict Stuxnet like advanced malware since it is the first advanced malware known in the wild. Additionally, conventional malware detection systems are unable to detect advanced malware and there is no dataset with enough advanced malware samples. Thus, we used malware datasets that contain traces of advanced malware similar to Stuxnet.

We analyzed the most common features of Stuxnet like advanced malware and then define the five features. We consider Windows API calls to defines malware features that we used to analyze malware. We used regression algorithms to test the proposed model on malware dataset. We applied linear, polynomial, and random forest regressions with 3 estimators. We also used R^2 score, Root mean square error (RMSE), Mean square error (MSE) and Mean absolute error (MAE) as error metrics in our analyses. In the experiments, accuracies of the training and test sets for linear and polynomial regression models are very close to each other therefore there is no need for hyper-parameters. Additionally, we do not consider effects of time and memory constraints in this paper. Briefly, we use correlations among the five features that we define to predict the type of malware.

We use Pearson correlation test to observe the linear relationship between five different malware properties in the property set. The value range of the correlation coefficient is between -1 and 1. Correlation equal to 1 implies that a linear equation shows the relationship between X and Y perfectly. On the other hand, correlation equal to -1 implies that all data points lie on a line for which Y decreases as X increases. A value of zero means that there is no linear correlation between variables.

4.1. Dataset

We prepare a dataset from two datasets to be able to increase the number of malware instances and the accuracy of predictions. We use malware samples to show correlations among malware features. The first dataset from which we collected malware instances was created during a data mining competition at the International Conference on Neural Information Processing in 2010. This dataset contains API calls and a label for each software. The label indicates the type of software, namely malware or benign. The dataset may be found in servers of Artificial Intelligence Laboratory at University of Arizona [39]. The second dataset consists of labeled software, where malware is defined with a name, API Calls, and an SHA256 value [40]. Since our goal with this analyses is to predict Stuxnet like advanced malware, we removed benign software from our dataset.

We intentionally do not specify types of malware on the dataset to be able to test our approach for the prediction of Stuxnet like advanced malware.

Our dataset consists of 23,759 malware instances. In the dataset, each line contains API calls that represents malware sample. The proposed approach needs training and test sets for the prediction of advanced malware. Therefore, at the beginning of analyses, we determine the outlier, training and test sets.

In experiments, we discovered that there are correlations among the five features. Moreover, we found that there is much more correlation between Stuxnet Clossness and Stealthiness features than correlations between other features. Specifically, we found that $A(Stealthiness, P) = 0.4$ and there is a correlation between $A(Stealthiness,\ P)$ and $A(StuxnetCloseness,\ P)$ as in Fig. 2. In the region where $A(Stealthiness,\ P) > 0.4$, there are only 9 malware instances. We treated them as outliers and then removed from our dataset.

We used %75 of the remaining malware instances as a training set and %25 of instances as a test set in the dataset. We used regression algorithms to model the correlations. We observe that there are statistically high linear correlations between two features. Therefore, we concentrated on simple models like linear regression. These models also prevent over fitting in our analyses.

4.2. Features

We determine five significant features of advanced malware that will be used to identify them after careful analyses of such malware samples in the wild. Moreover, we have concentrated on features of Stuxnet like advanced malware to have more precise predictions. Starting with features of specific advanced malware helps us to design more effective models for malware predictions and detections since there is a continuous race between malware creators and defenders.

In this paper, we only consider advanced malware traces on Windows platforms. Our features for malware predictions are based on Windows API calls. Specifically, the five features are stealth, conventional malware arsenal, Stuxnet closeness, behavioral instability, and metamorphic engine. We omit other properties and corresponding Windows API calls, which may be used by other malware detection approaches. Features and their related Windows API calls are as follows.

- Stealthiness: FindFirstFileA, FindNextFileA, GetProcAddress, LoadLibraryA, OpenProcess, Sleep [41].
- Conventional malware arsenal: ShowWindow, GetWindow, WriteFile, WinExec, ShellExecuteA, OpenProcess, VirtualAlloc [32].
- Stuxnet closeness: LoadLibraryW, LoadLibraryA, GetModuleHandle, GetProcAddress, VirtualAlloc, VirtualFree [42].
- Behavioral instability: WriteFile, CreateFileA, CreateFileW, CloseServiceHandle, FindFirstFileA, FindNextFileA, FindClose, SearchPathA, SearchPathW, RegOpenKeyA, RegCreateKeyA, RegCreateKeyExa, RegCreateKeyExw, RegCreateKeyW, RegSetValueExa, RegSetValueExw, RegCloseKey, DeleteFileA, DeleteFileW, GetFileAttributesA, GetFileAttributesW, GetFileAttributesExA, GetFileAttributesExW, GetFileInformationByHandle,

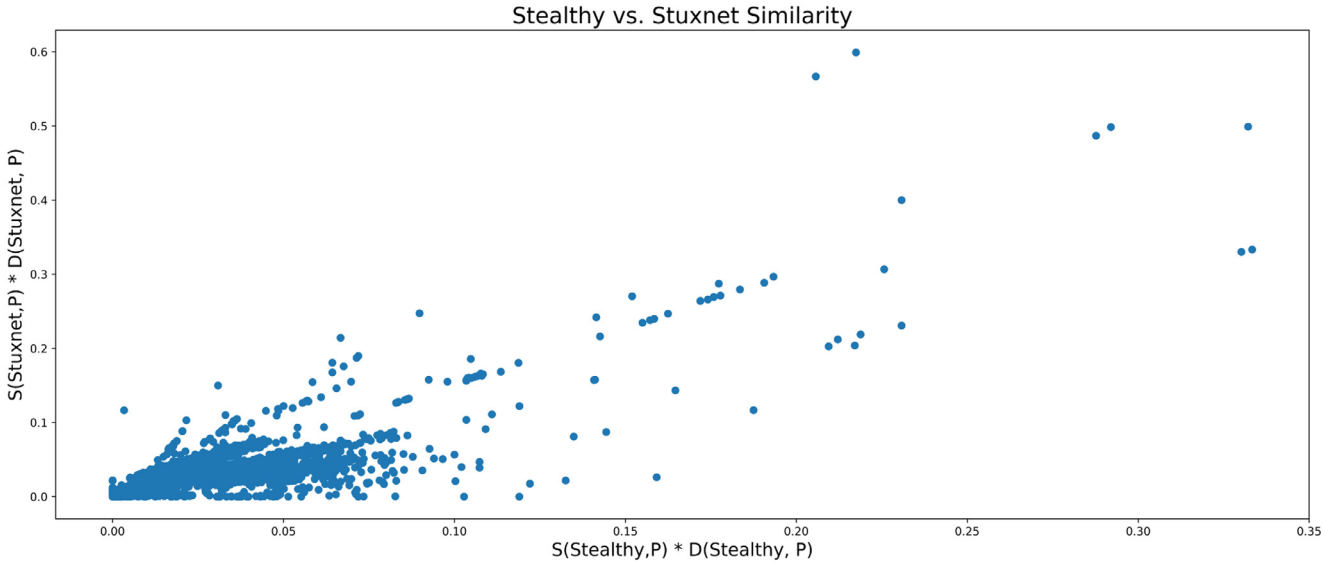


Fig. 2. The correlation between of Stuxnet closeness and stealthiness.

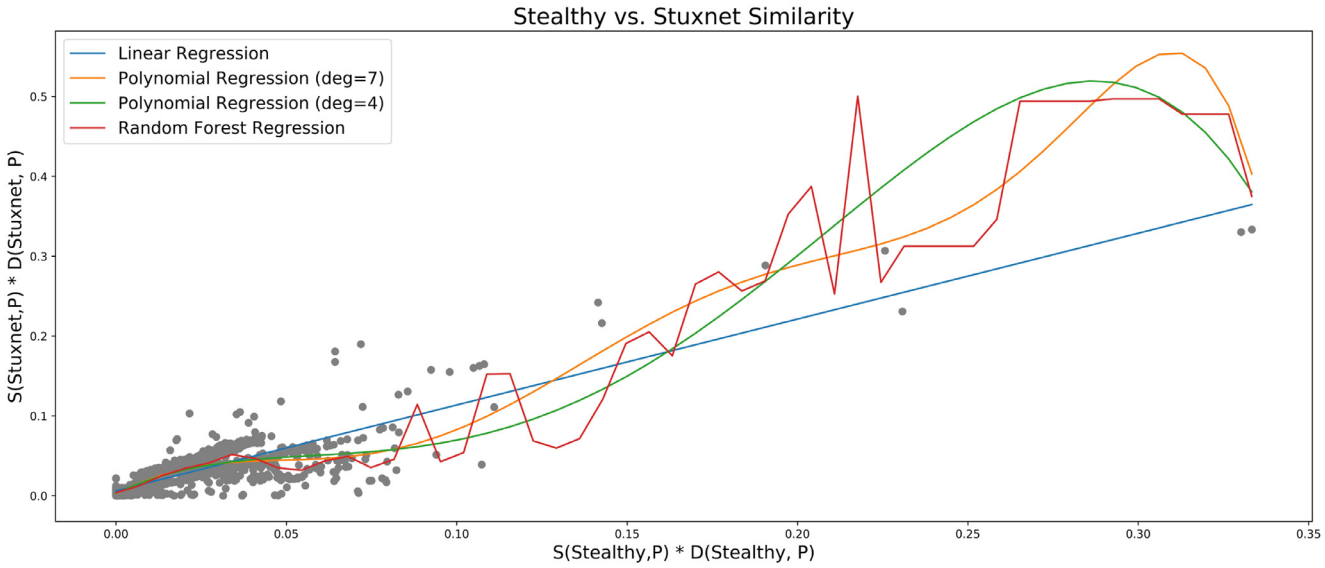


Fig. 3. Regression models on stealthiness & Stuxnet closeness.

GetFileSize, GetFileType, GetFullPathNameA, GetFullPathNameW, GetLongPathNameW, GetShortPathNameA, GetShortPathNameW, GetTempFileNameA, GetTempPathA, GetTempPathW, GlobalAlloc, GlobalFree, VirtualAlloc, VirtualFree, CopyFileA, DeleteFileA, DeleteFileW, GetFileSize, GetFileType, ReadFile [33,34].

- Metamorphic engine: HeapAlloc, LocalFree, HeapCreate, GetStartupInfoA, GetCommandLineA, GetEnvironmentStringsW, FreeEnvironmentStringsW, GetModuleFileNameA, GetCurrentProcess, CloseServiceHandle, GetCurrentProcessId, GetProcessHeap, HeapReAlloc, SetFilePointer, SetFileAttributesA, GetFileAttributesW, FindFirstFileA, FindClose, SetThreadPriority, GetCurrentThreadId, GetProcAddress, GetModuleHandleA, ResumeThread, GetEnvironmentVariableA, ExitThread [43].

Advanced malware has become more complex and it may have used new API calls related to the five features in new samples. In this paper, we focus on Stuxnet like advanced malware traces only on Windows platforms.

4.3. Evaluation of malware data with two features

We analyzed the correlations between any two features by using machine learning algorithms to extract potential relationships between the features to distinguish advanced malware and conventional malware. We applied three machine learning algorithms namely, linear regression, polynomial regression, and random forest regression on the dataset we constructed to extract correlations among the properties of malware. These correlations helps us to predict Stuxnet like advanced malware. The distribution of data in our dataset according to stealthiness and Stuxnet closeness features is shown in Fig. 2.

The results of selected machine learning algorithms that consider stealthy and closeness to Stuxnet features are shown Fig. 3. We use an outlier that filters the traces having more than 0.4 stealthy property to be able to handle correlations properly. The linear regression R^2 score is 0.7109 for the test data. R^2 scores for polynomial regressions with degree 4 and 7 are 0.7539 and 0.7759 respectively. On the other hand, Random forest regression with

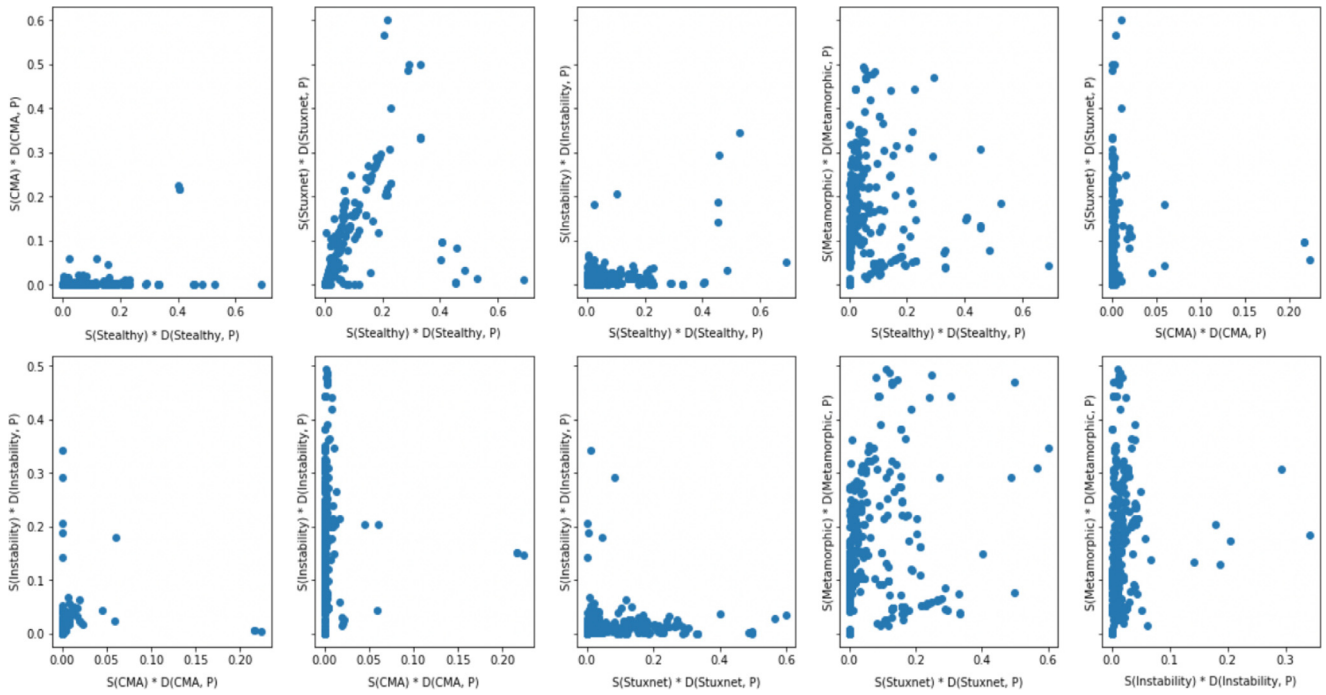


Fig. 4. Correlations between two properties of malware.

Table 2
Scores of regression algorithms.

Algorithm	R^2 scores	RMSE	MSE	MAE
Polynomial Regression (d = 7)	0.7759	0.0102	0.0001	0.0057
Polynomial Regression (d = 4)	0.7539	0.0107	0.0001	0.0059
Linear Regression	0.7109	0.0116	0.0001	0.0077
Random Forest Regression (tree = 3)	0.8208	0.0092	0.0001	0.0041

three estimators has 0.8208 R^2 score, which is greater than linear and polynomial regression scores. Table 2 contains scores of all machine learning algorithms applied to the dataset. Interestingly, MSE scores for all machine learning algorithms are smaller than other types of scores while error measures with R^2 score seems more meaningful in these experiments.

The experimental results show that stealthy and Stuxnet closeness properties are easily represented with linear and polynomial regressions. This means that there is no need for more complex machine learning algorithms to extract correlations between stealthy and Stuxnet closeness features. Less complex machine learning algorithms require smaller computation power with higher performance. The smallest number of Stuxnet like advanced malware instances and API calls related to them makes also the two features suitable for regression algorithms.

Fig. 4 shows all correlations between any two features. Analyses results show that correlations of Conventional Malware Arsenal (CMA) property with any other property differs from the correlation between stealthy and Stuxnet closeness properties. For example, the average score of stealthy feature of malware dataset is greater than the average score of CMA feature. Moreover, a similar distribution of closeness scores is between CMA and Stuxnet features. These experimental results show that if we know closeness of two features, we will use it to predict the type of malware, advanced or conventional. Moreover, if we know closeness for more than two features, such as Stuxnet closeness-stealthy and CMA-stealthy, we may predict the type of malware more accurately.

The number of API calls related to a feature affects the closeness score. CMA closeness score is relatively small than scores of other features. The main reason for this difference is that there are limited number of API calls related to this feature. On the other hand, there are much more API calls related to behavioral instability and metamorphic engine features. Therefore, features with more API calls may need more complex machine learning algorithms to have better closeness scores.

We extracted all one to one correlation scores of five features in the dataset as in Fig. 5. These scores show that there is a high linear correlation between stealthy score and Stuxnet closeness score since the correlation coefficient is 0.829, which is compatible with our previous analyses. Additionally, Stuxnet closeness and metamorphic engine score is the second higher score of Stuxnet closeness measures, which is 0.558. In malware literature, metamorphic engine is usually used to deceive anti-malware systems therefore it is related to stealthy property. The correlation score between metamorphic engine and stealthy features is a proof for this relationship. Thus, Stuxnet closeness, stealthy, and metamorphic engine features may be used to increase prediction accuracy of advanced malware.

All other correlation scores are below 0.5 in Fig. 5. These scores imply that there are low linear relationships between most of the properties. Correlations among stealthy, Stuxnet closeness, and metamorphic engine properties are notable for this dataset. These results show that extracting the correlations among features of advanced malware may help to predict specific types of advanced malware like Stuxnet.

Analyses results with two features show that there are different correlations between properties of malware. Therefore, specific machine learning algorithms should be applied to particular pairs of malware properties to predict advanced malware. Since there are a limited number of advanced malware instances in the wild, it is almost impossible to have enough data about all features of advanced malware. Thus, some correlations may be used to predict other correlations, so that correlations may be used to feed machine learning algorithms to predict advanced malware.

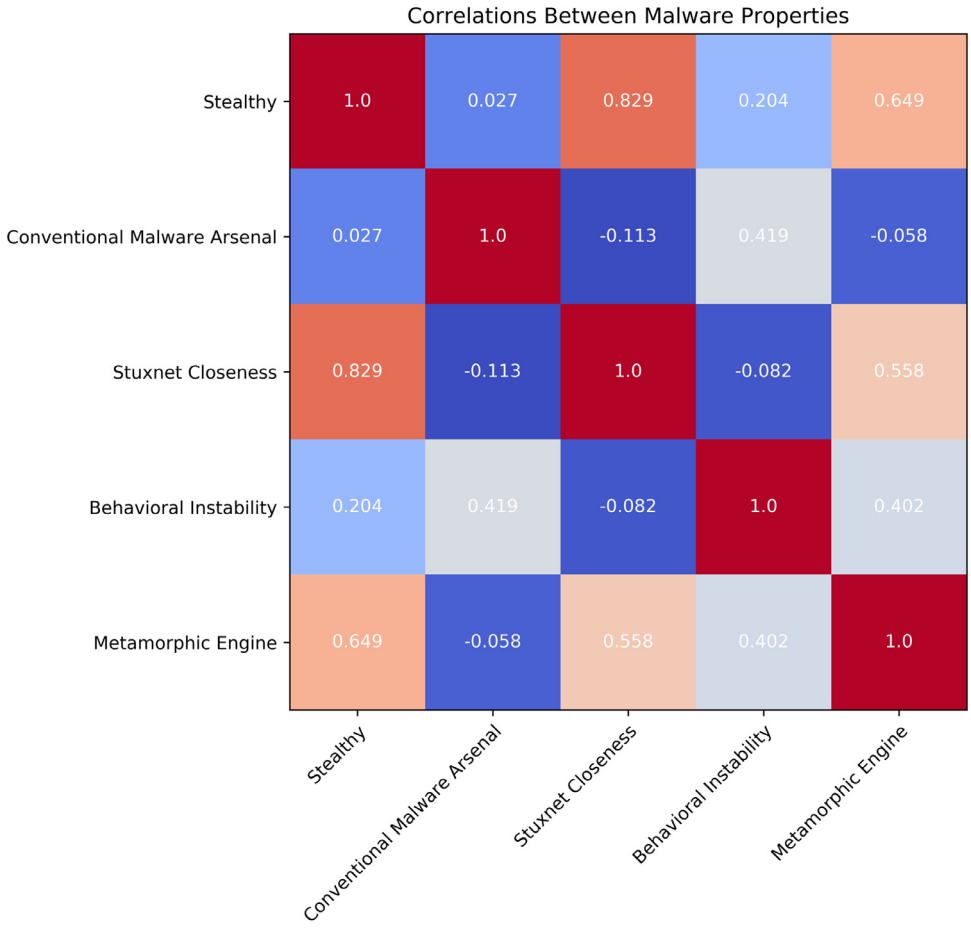


Fig. 5. All one to one correlations between features in the dataset.

4.4. Evaluation of malware data with three features

We analyzed all correlations between two features for the data set. We found that some features are more correlated than others. Moreover, analysis results with two features may be used to predict the correlations with other features, such as correlations with a third feature. Here, we analyzed correlations among three features to extract relationships among the properties more precisely. We concentrated on correlations among Stuxnet closeness and stealthy features with others since these two properties possess maximum similarity score in experiments of malware with two features.

The analyses results show that all features are somewhat dependent to other features. For instance, stealthy property affects almost all features considerably while we consider any other feature simultaneously. Particularly, Figs. 6 and 7 shows the effect of stealth feature on CMA, Stuxnet closeness, and behavioral instability features. In this case, the stealthy feature seems relatively dominant. On the other hand, the same feature has less effects on other features as in Figs. 8–10.

Some features have small effects on correlations in malware datasets. For example, CMA feature has less correlations with a third feature, such as the correlations among CMA, Behavioral Instability, and Metamorphic properties as shown in Figs. 11 and 12. These results are consistent with two feature analysis of malware.

Correlations among features show that advanced malware is created with a care. This means that API calls and properties are intentionally dependent to each other. Additionally, features and API calls have different level of dependencies. Therefore, correlations among the features should be extracted in more details. These correlations present the structure and the behavior of advanced malware, which may be used with machine learning algorithms to predict the type of malware.

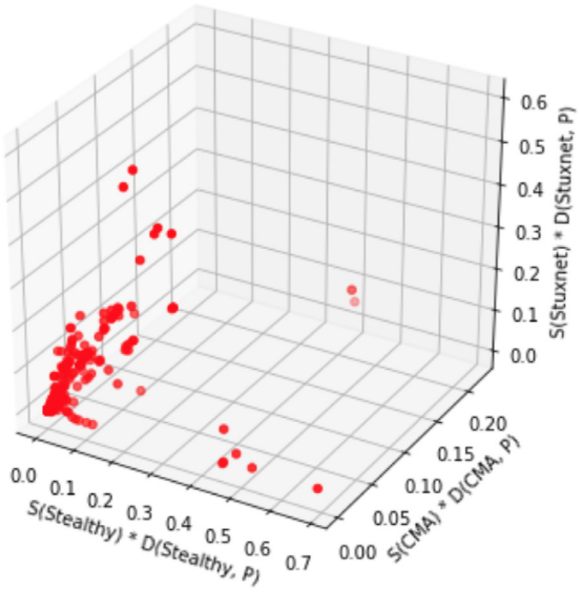


Fig. 6. Correlations among Stealthy, CMA, and Stuxnet Closeness features in the dataset.

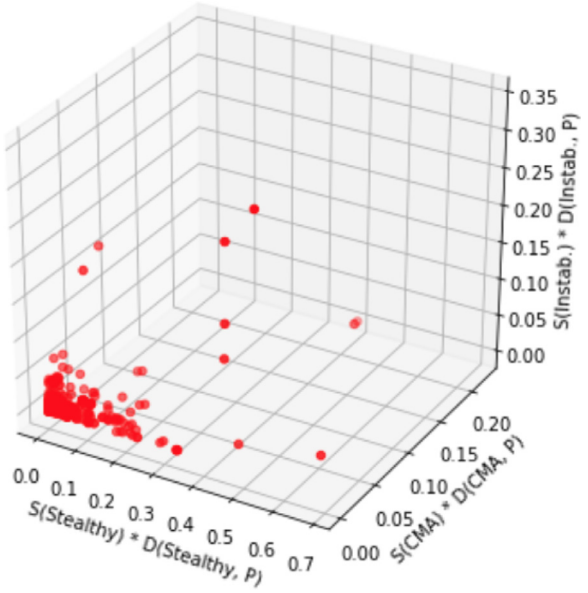


Fig. 7. Correlations among Stealthy, CMA, and Behavioral Instability features in the dataset.

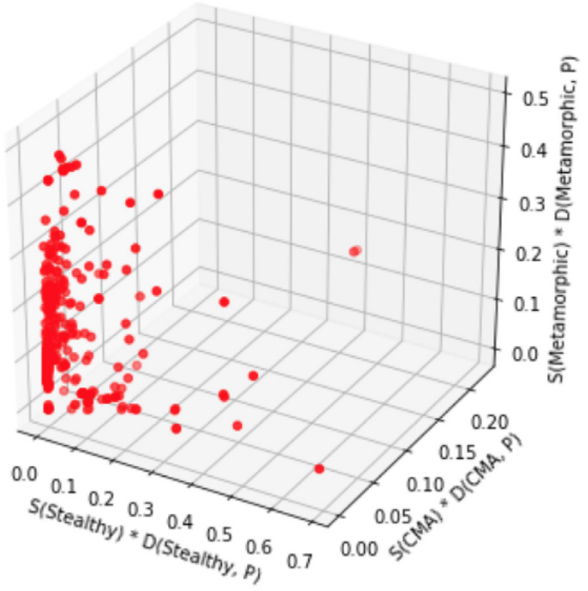


Fig. 8. Correlations among stealthy, CMA, and metamorphic features in the dataset.

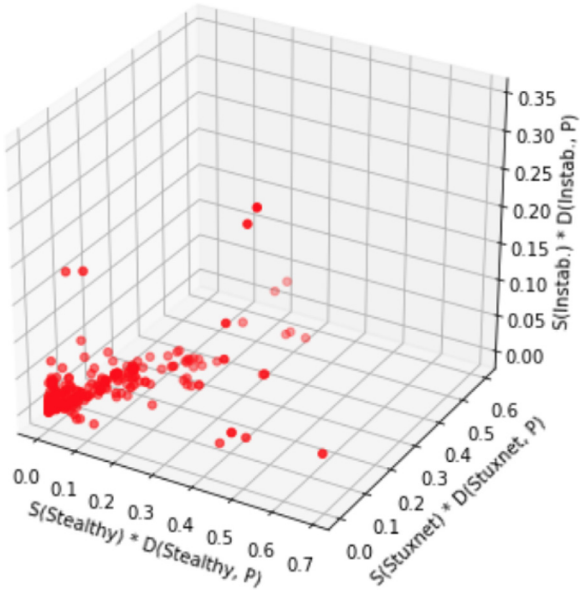


Fig. 9. Correlations among stealthy, Stuxnet closeness, and behavioral instability features in the dataset.

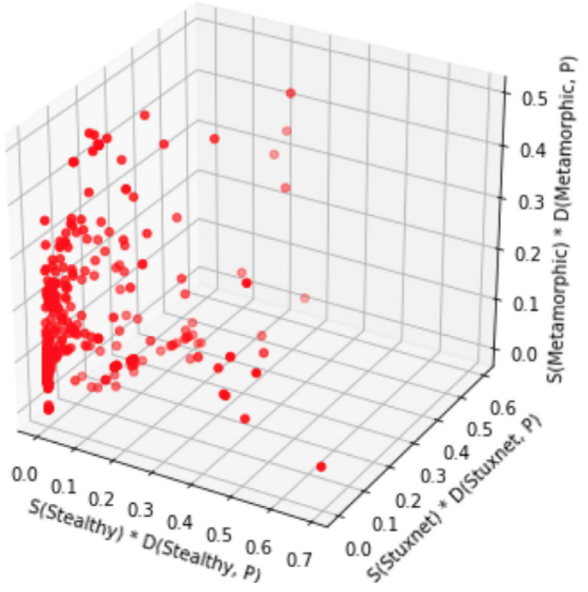


Fig. 10. Correlations among stealthy, Stuxnet closeness, and metamorphic features in the dataset.

4.5. Advanced malware prediction

We go into details to analyze the correlations among the five features. Since our goal in this paper is to predict Stuxnet like advanced malware and our dataset has the highest score for the correlation between stealthy and Stuxnet closeness features, here we used Random Forest Regressor models to predict the correlation of the two features of malware using other four properties. Specifically, we predict advanced malware if it has high similarity score to the correlations between stealthy and Stuxnet closeness by using other four features.

A potential method to distinguish conventional malware and advanced malware by using the score defined in this paper is to use a threshold. Let assume that the threshold is 0.8. If the score is above 0.8, then malware will be considered as advanced. Otherwise, it will be considered as conventional malware. Note that the exact value of the threshold for prediction of advanced malware depends on dataset and it is beyond the scope of this paper.

We found some promising predictions only for stealthy and Stuxnet closeness scores using other four properties. Moreover, analysis results show that Linear and Polynomial Regressions are inefficient this case. Therefore, we proceeded with Random Forest Regressions.

We used Random Forest Regression, where the number of trees is 3. The experimental result has $R^2 = 0.8203$ score for Stuxnet Closeness property to predict advanced malware. This score is higher than scores of other four models, where we only use two or three features to predict advanced malware. Specifically, experimental analysis with four features provide better scores to predict advanced malware. This means that the number of features used with Random Forest Regression is a significant issue to have more precise predications. Fig. 13 shows the prediction of Stuxnet Close-

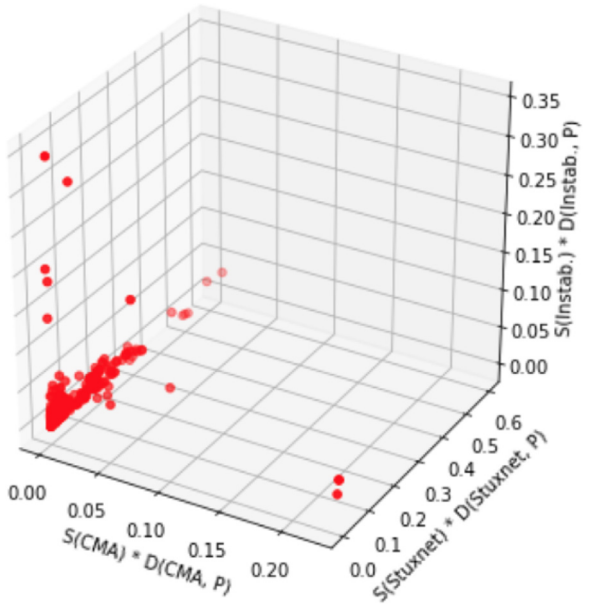


Fig. 11. Correlations among CMA, Stuxnet closeness, and behavioral instability features in the dataset.

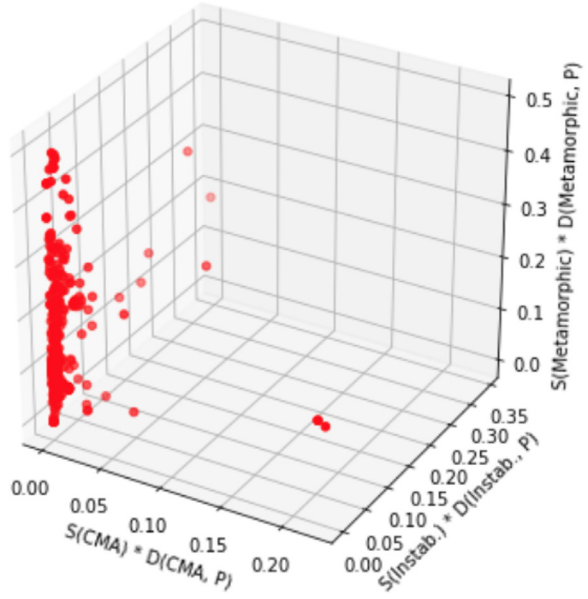


Fig. 12. Correlations among CMA, Behavioral instability, and metamorphic features in the dataset.

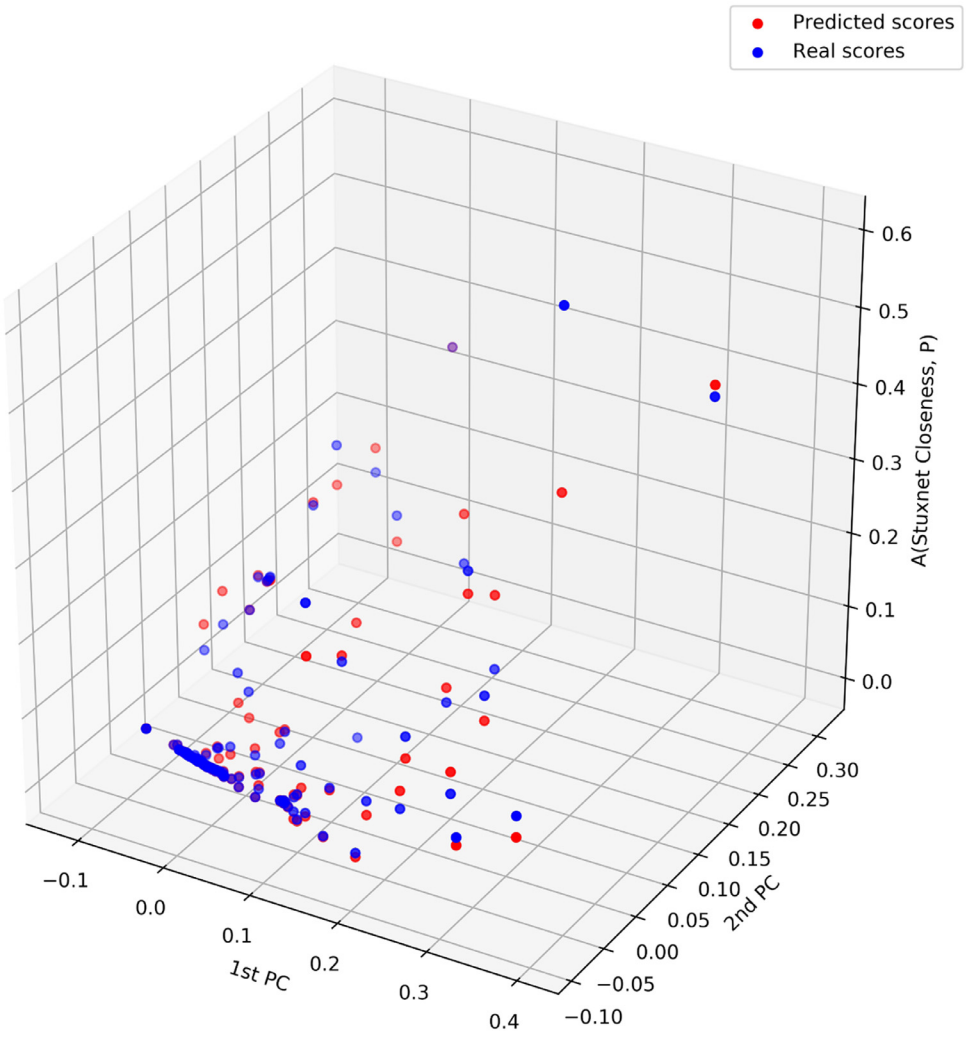


Fig. 13. Prediction of Stuxnet closeness scores using random forest regressor with four properties as features.

ness using Random Forest Regression with the four features. In the figure, PC means a principle component. For the sake of brevity, we reduced the 4-dimensional data to 2-dimensional. Then, we visualized the predictions in 3-dimensional. In this analysis, our goal is to predict Stuxnet closeness score by using other four features. If the score is greater than the threshold, which is used to distinguish conventional malware and Stuxnet like advanced malware, then malware is advanced. If we consider experimental results given in Fig. 13, the value of 0.3 for the threshold will be meaningful to predict advanced malware, where there are three real malware with two acceptable predictions.

Analyses results of analysis have two important consequences. Some machine learning models provide better prediction results for features while the complexity of predictions increases, such as Random Forest Regression model then linear regression and polynomial regression models. Second significant result observed is that properties of malware have dependencies to each other with different scores. Thus, specific machine learning algorithms should be used with particular features to predict advanced malware. Another significant issue is to determine acceptable threshold to identify advanced malware, which is beyond the scope of this paper.

5. Conclusion and future work

The volume and complexity of cyber-attacks that are carried out with malware become prevalent on almost all computing systems. Existing intrusion detection and anti-malware systems are unable to detect all attacks and malware on networks. Specifically, attacks with advanced malware are still undetectable with existing mechanisms. This brings a huge amount of additional cost to computing systems. There is an urgent need for mechanisms to predict advanced malware that will be used to detect the attacks.

This paper contains a brief explanation of conventional malware types to make clear their tasks. We analyzed Stuxnet like advanced malware instances in the wild since it has distinguishing properties than conventional malware. We found that some features of advanced malware are more correlated with some others, which may be used to identify such malware and corresponding attacks. Additionally, the paper contains an overview about machine learning based malware predictions and detections. We propose a multi-dimensional machine learning approach to predict Stuxnet like advanced malware by using five features. This paper contains three main contributions.

- After careful analyses of Stuxnet like advanced malware in the wild, we define five features for prediction purposes. These consist of API calls and features are defined only for Windows platforms.
- We present a machine learning approach to predict advanced malware. The proposed approach uses correlations among five features to predict such malware.
- The proposed approach is able to predict Stuxnet like advanced malware by using four regression algorithms with different number of features on a malware dataset. To date this is the first approach to predict Stuxnet like advanced malware by a dataset that consists of only malware samples.

Through our experimental results we have shown that features depend on each other with different correlation scores. The highest correlation is between stealthy and Stuxnet closeness features with a value 0.829. Stuxnet closeness and metamorphic engine score is 0.558. These results show that stealthy feature is highly related to metamorphic feature. On the other hand, some features have less correlations with others, such as CMA feature. For example, the correlation among CMA, behavioral instability, and metamorphic engine features has a minimum score. We observed that linear and polynomial regression algorithms are inefficient with four features while random forest regressions provide better scores with more features.

Since there are limited information about advanced malware instances in the wild, we have been working to tune our machine learning approach based on newly found advanced malware instances. For example, we arbitrarily selected the value of the threshold for predicting advanced malware. We are working to find a meaningful threshold with newly found malware samples.

Declaration of Competing Interest

None

Acknowledgment

This work is supported by Istanbul Teknik Üniversitesi under the BAP project, number MAB-2017-40642.

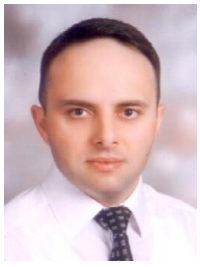
Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.comnet.2019.06.015.

References

[1] W. Yan, CAS: a framework of online detecting advance malware families for cloud-based security, in: Communications in China (ICCC), 2012 1st IEEE International Conference on, IEEE, 2012, pp. 220–225.
[2] B.A.S. Al-rimy, M.A. Maarof, S.Z.M. Shaid, Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions, Comput. Secur. 74 (2018) 144–166, doi:10.1016/j.cose.2018.01.001.
[3] J. Jansen, R. Leukfeldt, Phishing and malware attacks on online banking customers in the netherlands: a qualitative analysis of factors leading to victimization, Int. J. Cyber Criminol. 10 (1) (2016) 79.
[4] N. Kshetri, J. Voas, Banking on availability, Computer 50 (1) (2017) 76–80.
[5] Ş Bahtiyar, Anatomy of targeted attacks with smart malware, Secur. Commun. Netw. 9 (18) (2016) 6215–6226.
[6] V. Paxson, Viruses and Worms, 2011.
[7] T. Yagi, N. Tanimoto, T. Hariu, M. Itoh, Investigation and analysis of malware on websites, in: Web Systems Evolution (WSE), 2010 12th IEEE International Symposium on, IEEE, 2010, pp. 73–81.
[8] I. Zelinka, S. Das, L. Sikora, R. Senkerik, Swarm virus - next-generation virus and antivirus paradigm? Swarm Evolut. Comput. 43 (2018) 207–224, doi:10.1016/j.swevo.2018.05.003.
[9] CERT-UK, An Introduction to Malware, 2014. http://www.cert.gov.uk/resources/best-practices/an-introduction-to-malware/.
[10] M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, F. Palmieri, A. Castiglione, Using screen brightness to improve security in mobile social network access, IEEE Trans. Depend. Sec. Comput. 15 (4) (2018) 621–632, doi:10.1109/TDSC.2016.2601603.
[11] N. Zhang, R. Zhang, K. Sun, W. Lou, Y.T. Hou, S. Jajodia, Memory forensic challenges under misused architectural features, IEEE Trans. Inf. Forensics Secur. 13 (9) (2018) 2345–2358, doi:10.1109/TIFS.2018.2819119.
[12] D. Plohmann, E. Gerhards-Padilla, F. Leder, Botnets: measurement, detection, disinfection and defence, ENISA workshop on, 2011.
[13] E. Skoudis, L. Zeltser, Malware: Fighting Malicious Code, Prentice Hall Professional, 2004.
[14] N. Virvilis, D. Gritzalis, The big four-what we did wrong in advanced persistent threat detection? in: Availability, Reliability and Security (ARES), 2013 Eighth International Conference on, IEEE, 2013, pp. 248–254.
[15] A. Clark, Q. Zhu, R. Poovendran, T. Başar, An impact-aware defense against Stuxnet, in: American Control Conference (ACC), 2013, IEEE, 2013, pp. 4140–4147.
[16] G. Bonfante, J.Y. Marion, F. Sabatier, A. Thierry, Analysis and diversion of Duqu's driver, in: Malicious and Unwanted Software: "The Americas"(MALWARE), 2013 8th International Conference on, IEEE, 2013, pp. 109–115.
[17] N. Virvilis, D. Gritzalis, T. Apostolopoulos, Trusted computing vs. advanced persistent threats: can a defender win this game? in: Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), IEEE, 2013, pp. 396–403.
[18] I. Rosenberg, G. Sicard, E.O. David, End-to-end deep neural networks and transfer learning for automatic analysis of nation-state malware, Entropy 20 (5) (2018) 390, doi:10.3390/e20050390.
[19] W. Tounsi, H. Rais, A survey on technical threat intelligence in the age of sophisticated cyber attacks, Comput. Secur. 72 (2018) 212–233, doi:10.1016/j.cose.2017.09.001.

[20] A. Lemay, J. Calvet, F. Menet, J.M. Fernandez, Survey of publicly available reports on advanced persistent threat actors, Comput. Secur. 72 (2018) 26–59, doi:10.1016/j.cose.2017.08.005.
[21] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, V.C.M. Leung, A survey on security threats and defensive techniques of machine learning: a data driven view, IEEE Access 6 (2018) 12103–12117, doi:10.1109/ACCESS.2018.2805680.
[22] A. Souri, R. Hosseini, A state-of-the-art survey of malware detection approaches using data mining techniques, Hum. Centric Comput. Inf. Sci. 8 (2018) 3, doi:10.1186/s13673-018-0125-x.
[23] P. Burnap, R. French, F. Turner, K. Jones, Malware classification using self organising feature maps and machine activity data, Comput. Secur. 73 (2018) 399–410, doi:10.1016/j.cose.2017.11.016.
[24] S. Kim, J. Kim, S. Nam, D. Kim, Webmon: ML- and yara-based malicious webpage detection, Comput. Netw. 137 (2018) 119–131, doi:10.1016/j.comnet.2018.03.006.
[25] D. Ucci, L. Aniello, R. Baldoni, Survey of machine learning techniques for malware analysis, Comput. Secur. 81 (2019) 123–147, doi:10.1016/j.cose.2018.11.001.
[26] W. Han, J. Xue, Y. Wang, Z. Liu, Z. Kong, Malinsight: a systematic profiling based malware detection framework, J. Netw. Comput. Appl. 125 (2019) 236–250, doi:10.1016/j.jnca.2018.10.022.
[27] M.F.A. Razak, N.B. Anuar, R. Salleh, A. Firdaus, M. Faiz, H.S. Alamri, "less give more": evaluate and zoning android applications, Measurement 133 (2019) 396–411.
[28] T. Kim, B. Kang, M. Rho, S. Sezer, E.G. Im, A multimodal deep learning method for android malware detection using various features, IEEE Trans. Inf. Forensics Secur. 14 (3) (2019) 773–788, doi:10.1109/TIFS.2018.2866319.
[29] Q. Le, O. Boydell, B.M. Namee, M. Scanlon, Deep learning at the shallow end: malware classification for non-domain experts, Digital Invest. 26 (2018) S118–S126, doi:10.1016/j.diin.2018.04.024.
[30] P. Black, I. Gondal, R. Layton, A survey of similarities in banking malware behaviours, Comput. Secur. 77 (2018) 756–772, doi:10.1016/j.cose.2017.09.013.
[31] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K.M. Rabie, F.J. Aparicio-Navarro, Detection of advanced persistent threat using machine-learning correlation analysis, Fut. Gener. Comput. Syst. 89 (2018) 349–359, doi:10.1016/j.future.2018.06.055.
[32] S. Gupta, H. Sharma, S. Kaur, Malware characterization using windows API call sequences, in: International Conference on Security, Privacy, and Applied Cryptography Engineering, Springer, 2016, pp. 271–280.
[33] C. Wang, J. Pang, R. Zhao, W. Fu, X. Liu, Malware detection based on suspicious behavior identification, in: Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on, Vol. 2, IEEE, 2009, pp. 198–202.
[34] M. Alazab, S. Venkataraman, P. Watters, Towards understanding malware behaviour by the extraction of API calls, in: Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second, IEEE, 2010, pp. 52–59.
[35] B. Yu, Y. Fang, Q. Yang, Y. Tang, L. Liu, A survey of malware behavior description and analysis, Front. Inf. Technol. Electron.Eng. 19 (5) (2018) 583–603, doi:10.1631/FITEE.1601745.
[36] A. Firdaus, N.B. Anuar, M.F.A. Razak, I.A.T. Hashem, S. Bachok, A.K. Sangaiah, Root exploit detection and features optimization: mobile device and blockchain based medical data management, J. Med. Syst. 42 (6) (2018), doi:10.1007/s10916-018-0966-x. 112:1–112:23.
[37] M.H. Nguyen, D.L. Nguyen, X.M. Nguyen, T.T. Quan, Auto-detection of sophisticated malware using lazy-binding control flow graph and deep learning, Comput. Secur. 76 (2018) 128–155, doi:10.1016/j.cose.2018.02.006.
[38] M. Rhode, P. Burnap, K. Jones, Early-stage malware prediction using recurrent neural networks, Comput. Secur. 77 (2018) 578–594.
[39] CSDMC2010, Malware API Sequence Dataset, University of Arizona Artificial Intelligence Lab, AZSecure-data, 2017. Available http://www.azsecure-data.org/other-data.html.
[40] Y. Ki, E. Kim, H.K. Kim, A novel approach to detect malware based on API call sequence analysis, Int. J. Distrib. Sensor Netw. 11 (2015) 659101, doi:10.1155/2015/659101. 1–659101:9.
[41] E.M. Rudd, A. Rozsa, M. Günther, T.E. Boult, A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions, IEEE Commun. Surv. Tut. 19 (2) (2017) 1145–1172.
[42] A. Matrosov, E. Rodionov, D. Harley, J. Malcho, Stuxnet Under the Microscope, ESET LLC, 2010.
[43] V.P. Nair, H. Jain, Y.K. Golecha, M.S. Gaur, V. Laxmi, Medusa: metamorphic malware dynamic analysis usingsignature from API, in: Proceedings of the 3rd International Conference on Security of Information and Networks, ACM, 2010, pp. 263–269.



Şerif Bahtiyar received his B.S. degree in control and computer engineering and M.S. degree in computer engineering from İstanbul Technical University, Turkey and his Ph.D. degree in computer engineering from Boğaziçi University, İstanbul, Turkey. Dr. Bahtiyar was a senior researcher at National Research Institute of Electronics and Cryptology, Turkey. He was a post-doctoral researcher at TU-Berlin, Germany. Dr. Bahtiyar served as a product manager at MasterCard and he joined the Department of Computer Engineering of İstanbul Technical University, Turkey, where he is currently serving as an assistant professor. His current interests are security, trust, financial systems, smart systems, and product management.