Souri and Hosseini *Hum. Cent. Comput. Inf. Sci.* (2018) 8:3
https://doi.org/10.1186/s13673-018-0125-x

Human-centric Computing and Information Sciences

RESEARCH Open Access

A state-of-the-art survey of malware detection approaches using data mining techniques

Alireza Souri[1*] and Rahil Hosseini[2]

*Correspondence:
a.souri@srbiau.ac.ir
[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
Full list of author information is available at the end of the article

Abstract

Data mining techniques have been concentrated for malware detection in the recent decade. The battle between security analyzers and malware scholars is everlasting as innovation grows. The proposed methodologies are not adequate while evolutionary and complex nature of malware is changing quickly and therefore turn out to be harder to recognize. This paper presents a systematic and detailed survey of the malware detection mechanisms using data mining techniques. In addition, it classifies the malware detection approaches in two main categories including signature-based methods and behavior-based detection. The main contributions of this paper are: (1) providing a summary of the current challenges related to the malware detection approaches in data mining, (2) presenting a systematic and categorized overview of the current approaches to machine learning mechanisms, (3) exploring the structure of the significant methods in the malware detection approach and (4) discussing the important factors of classification malware approaches in the data mining. The detection approaches have been compared with each other according to their importance factors. The advantages and disadvantages of them were discussed in terms of data mining models, their evaluation method and their proficiency. This survey helps researchers to have a general comprehension of the malware detection field and for specialists to do consequent examinations.

Keywords: Data mining, Malware detection, Classification, Behavior-based, Signature-based

Introduction

In the recent years, the application of malware detection mechanisms utilize through data mining techniques through have increased using machine learning to recognize malicious files [1, 2]. Machine learning methods can take in hidden examples from a given preparing set which includes both malware and benign examples. These basic examples can separate malware from benevolent code [3, 4]. Malware is a standout most thoughtful intimidations for distributed systems and the Internet [5]. The battle between security analyzers and malware scholars is everlasting as innovation grows. Malware is a program that makes your framework accomplish something that an assailant needs it to do [6]. The most generally utilized malware detection develops a straightforward example coordinating way to deal with identify vindictive code. Typically malware designers

Springer Open

© The Author(s) 2018. This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

don't compose new code without any preparation, yet redesign the old code with new components or muddling strategies [7]. With a large number of malware cases seeming each day, proficiently preparing countless specimens which display comparable conduct, has turned out to be progressively essential [8].

Up to now, malware analysis [9, 10] have the high growing impact in the procedure of deciding the reason and the usefulness the conduct of a given suspicious application. Such a procedure is an important essential with a specific end goal to create effective and powerful identification furthermore characterization techniques; malware analysis is partitioned into two primary classifications that include dynamic and static methods [11, 12]. To the best of our knowledge, the most data mining methods have some benefits and weaknesses in malware detection subject [13]. In addition, having a new literature review can be influenced on the research studies and explore some technical details in malware detection using data mining techniques. Of course, some research [13–17] had discussed the malware detection approaches. There are some defects in the surveyed research. Some papers are published in out of date and did not considered new articles in comparison and analysis. In addition, some surveys have not any systematic classification and article selection for their researches. For example, Muazzam Siddiqui et al. [18] presented a survey of malware detection using data mining techniques. Some defects of the survey are as follow: this survey used old research in literature analysis. In addition, they did not any systematic review for article selection in their research. This research did not specified an appropriate categorization for malware detection techniques. Just, they analyzed the scanning and data analysis methods in the proposed research.

To overcome some defects, this paper presents a systematic literature review on the new recent malware detection techniques using data mining approaches. This review classifies the malware detection approaches in two main fields: signature-based and behavior-based. The contributions of this paper are as follows:

- Providing a summary of the current challenges related to malware detection approaches in data mining.
- Presenting a systematic and categorized overview of the current approaches to machine learning mechanisms in the data mining topics.
- Exploring a structure of the important methods that are significant in malware detection approach.
- Discussing the important factors of classification malware approaches in the data mining to improve their problems in the futures.

The rest of this paper organized as follows. "Malware detection approaches", overviews the malware detection mechanisms in data mining methods and classifies them with a technical taxonomy. "Review of the malware detection approaches" presents an analytical comparison of the proposed approaches for selected studies. In "Discussion", a discussion about the malware detection issues is shown that have not been analyzed comprehensively up to now as an exploration of new challenges. Finally, "Conclusion" displays the conclusion.

Malware detection approaches

As a result of the developing malware in the innovation, the information of obscure malware protection is a fundamental subject in the malware recognition as per the machine learning strategies [19]. The machine learning strategies are divided into supervised and unsupervised classes. Malware detection approaches are divided into two main categories that include behavior-based and signature-based methods [20]. Also, there are two static and dynamic [21] malware analysis that generally performed in finding malicious applications [22].

In Fig. 1, we illustrate a malware detection taxonomy based on machine learning approaches. According to this figure, the API calls features, assembly features, and binary features are existing approaches for malware detection method. These features use machine learning methods for predicting and detecting malicious files.

Signature-based malware detection

Recently, signature-based detection is the most generally utilized procedure in antivirus programming highlighting exact correlation. Malware recognition has essentially centered on performing static investigations to review the code-structure mark of infections, instead of element behavioral methods [23]. The signature-based system finds interruptions utilizing a predefined list of known assaults. Despite the fact that this arrangement has the ability to identify malware in the versatile application, it requires steady overhauling of the predefined signature database. Moreover, it is less effective in identifying noxious exercises utilizing the signature-based technique because of the quickly changing nature of portable malware [24, 25]. Signature-based strategies depend in light of exceptional crude byte examples or standard articulations, known as marks, made to coordinate the noxious document. For example, static highlights of a record are utilized to decide if it is a malware. The main advantage of signature-based techniques is their thoroughness since they follow all conceivable execution ways of a given document.

In inside of the malware structure, existing malicious objects have characteristics that can be used to generate a unique digital signature. The anti-malware provider utilizes the meta-heuristic algorithms that can scan efficiently the malicious object to control its signature [26]. After identifying the malicious object, the detected signature is added

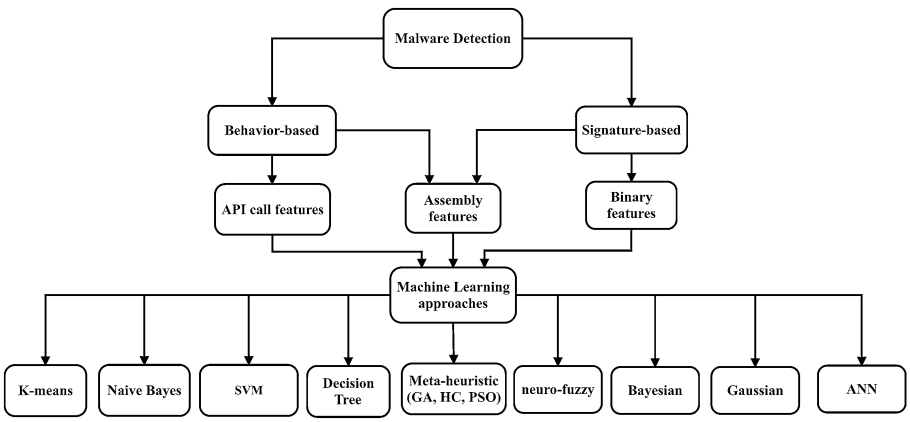

Fig. 1 Taxonomy of malware detection approaches

to the existing database as the recognized malware. The database sources include huge number of the various signatures that classify malicious objects. In the signature-based malware detection, there are some various qualities including fast identification, easy to run, and broadly accessible [27].

Since the digital signature plans are gotten from known malware, these plans are likewise generally known. Subsequently they can be effectively evaded by programmers utilizing straightforward confusion procedures. Hence malware code can be modified and signature-based identification can be sidestepped. Since anti-malware providers are built on the premise of known malware, they can't to distinguish obscure malware, or even variations of known malware. In this way, without exact digital signature, they can't adequately distinguish polymorphic malware. Along these lines, signature-based recognition does not give zero-day insurance. Besides, since a signature-based indicator utilizes an isolate signature for each malware variation, the database of signatures develops at an exponential rate [28]. The signature-based malware detection has two main methods for applying malware detection approach in machine learning methods including assembly features and binary features. Figure 2 illustrates a standard signature-based malware detection framework using data mining approaches.

Also, Table 1 shows the advantages and weaknesses of the signature-based malware detection approach.

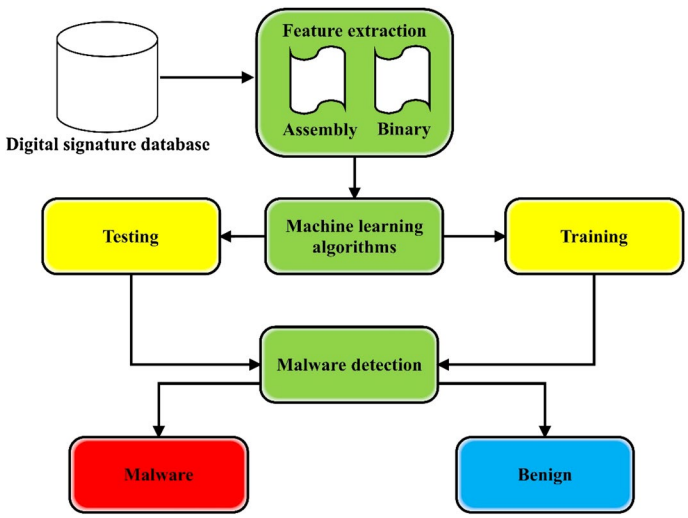

Fig. 2 The signature-based malware detection framework

Table 1 The advantages and weaknesses of the signature-based detection

Advantage	Weakness
Easy to run	Failing to detect the polymorphic malwares
Fast identification	Replicating information in the huge database
Broadly accessible	
Finding comprehensive malware information	

Behavior-based malware detection

This subsection illustrates the behavior-based approaches in malware detection. In addition, it reviews the selected behavior-based approaches in the data mining. Finally, the discussed behavior-based approaches compared and summarized in the last subsection. Behavior-based methodologies require execution of a given example in a sandboxed situation and run-time exercises are checked and logged. Dynamic investigation systems utilize both virtualization and imitating conditions to execute a malware and to remove its practices. The primary advantage of the behavior-based approach is that gives a superior comprehension of how malware is produced and implemented [8, 14].

In the behavior-based malware approach, the suspicious objects are assessed based on their activities that they cannot execute in system. Efforts to achieve activities that are clearly irregular or unofficial would specify the suspicious object is malicious, or at least apprehensive. A malicious behavior is known using a dynamic analysis that evaluates malicious intent by the object's code and structure. In the behavior-based detection, the API calls and assembly features are two main methods for applying machine learning algorithms. Figure 3 depicts a standard behavior-based malware detection approach using data mining algorithms.

Table 2 shows the advantages and weaknesses of the behavior-based malware detection approach.

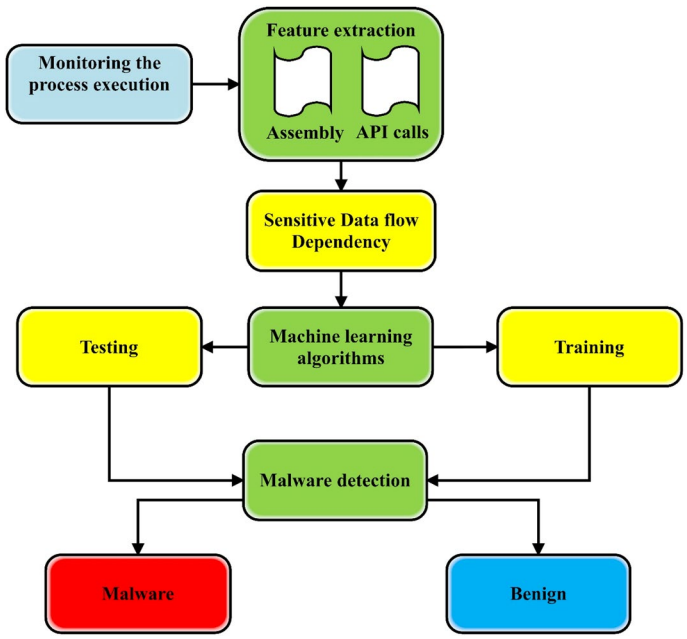

Fig. 3 The behavior-based malware detection framework

Table 2 The advantages and weaknesses of the behavior-based detection

Advantage	Weakness
Detecting unconceived types of malware attacks	Storage complexity for behavioral patterns
Data-flow dependency detector	Time complexity
Detecting the polymorphic malwares	

After describing the existing malware detection approaches, next section presents the technical analysis of the current research studies in the malware detection with data mining algorithms.

Review of the malware detection approaches

In this section, the existing malware detection approaches are analyzed according to some evaluation factors such as the main idea, advantages and disadvantages, algorithm type and assessment type in data mining techniques. We analyze the selected studies according to existing approaches and discuss on them.

Review of the signature-based approaches

Wu et al. [29] have utilized an artificial immune-based smartphone malware detection model (SP-MDM) both static malware examination and element malware investigation as indicated by the component of the biologic resistant framework that can shield us from disease by creatures. In this model, the static marks and dynamic marks of malware are separated, and in view of the genuine esteemed vector encoding, the antigens are produced. The youthful identifier develops into a develop one on the off chance that it experiences self-resistance. Finder posterity with higher fondness is made after the streamlining of developing identifiers utilizing clonal determination calculation. Also, they collected twenty malware and twenty benign files as testing samples set.

Bat-Erdene et al. [30] presented a strategy for characterizing the packing algorithms of given unknown packed executable. To begin with, they measured the entropy estimations of a given executable and change over the entropy estimations of a specific area of memory into typical representations. Their presented strategy utilized symbolic aggregate approximation (SAX), which is known to be viable for huge information changes. Second, we order the conveyance of images utilizing managed learning order strategies, i.e., credulous Bayes and bolster vector machines for recognizing pressing calculations. The aftereffects of our examinations including a gathering of 324 pressed kindhearted projects and 326 stuffed malware programs with 19 pressing calculations illustrate that our strategy can distinguish pressing calculations of given executable with a high precision of 95.35%, a review of 95.83%, and an accuracy of 94.13%. We propose four likeness estimations for distinguishing pressing calculations based on SAX representations of the entropy values and an incremental total examination. Among these four measurements, the loyalty closeness estimation shows the best-matching result, i.e., a rate of precision running from 95.0 to 99.9%, which is from 2 to 13 higher than that of the other three measurements. Our review affirms that pressing calculations can be recognized through an entropy examination in view of a measure of the instability of the running procedures and without earlier information of the executable.

Cui et al. [31] illustrated a novel recognition framework in light of cloud environment and packet examination. The framework identifies the malicious mobile malware behavior through their bundles with the utilization of information mining strategies. This approach totally keeps away from the deformities of customary techniques. The framework is administration arranged and can be sent by portable administrators to send cautions to clients who have malware on their gadgets. To enhance framework execution, another bunching technique called withdrawal grouping was made. This technique

utilizes earlier learning to lessen dataset measure. In addition, a multi-module location plan was acquainted with improve framework precision. The aftereffects of this plan are created by incorporating the location consequences of a few calculations, including Naive Bayes and Decision Tree.

Fan et al. [32] proposed a compelling arrangement mining calculation to find vindictive quintal examples, and afterward, All-Nearest-Neighbor (ANN) classifier is constructed for malicious position in the established samples. The created information mining structure made out of the proposed consecutive example mining technique and ANN classifier can well describe the malevolent examples from the gathered record test set to adequately distinguish recently concealed malware tests. A thorough exploratory review on a genuine information accumulation is performed to assess our recognition structure. The promising test comes about demonstrate that their structure beats other to exchange information mining based discovery techniques in distinguishing new vindictive executable.

Hellal and Ben Romdhane [33] displayed another diagram mining technique to recognize variations of malware utilizing static examination while covering the current defects. Also, they proposed a novel calculation, called minimal contrast frequent subgraph miner method (MCFSM), for separating negligible discriminative and generally utilized malevolent behavioral designs which can distinguish definitely a whole group of vindictive projects, conversely to another arrangement of benevolent projects. The proposed technique demonstrates high recognition rates and low false positive rates and creates a predetermined number of behavioral malware marks.

Martín et al. [34] illustrated outsider calls to sidestep the impacts of these disguise methodologies since they can't be obfuscated. We join bunching and multi-target advancement to produce a classifier in view of particular practices characterized by outsider call bunches. The analyzer guarantees that these gatherings are identified with noxious or favorable practices cleaning any non-discriminative example. This device, named MOCDroid,[1] accomplishes a precision of 95.15% in test with 1.69% of false positives with genuine applications extricated from the wild, overcoming all business antivirus motors from VirusTotal.

Santos et al. [35] proposed another strategy to identify obscure malware families. This model depends on the recurrence of the presence of opcode groupings. Moreover, they depicted a system to mine the importance of each opcode and evaluate the recurrence of each opcode grouping. Furthermore, they provided experimental approval that this new strategy is fit for recognizing obscure malware.

Wang and Wang [24] presented a malware recognition framework to ensure a little order mistake by machine learning using the speculation capacity of support vector models (SVMs). This review built up a programmed malware location framework via preparing a SVM classifier in light of behavioral marks. Over approval, plan was utilized for taking care of grouping exactness issues by utilizing SVMs connected with 60 groups of genuine malware. The trial comes about uncover that the characterization blunder diminishes as the measuring of testing information is expanded. For various estimating (N) of malware tests, the expectation precision of malware discovery runs up to 98.7%

[1] Multi-objective classifier detection.

with N = 100. The general recognition precision of the SVC is more than 85% for unspecific versatile malware.

Summary of the reviewed signature-based approaches

According to the discussed and reviewed signature-based detection approaches, the comparison of the proposed articles is demonstrated in Table 3 which shows the used case study in research, the main advantages, disadvantages and target environment for the existing studies. The main advantage of signature-based detection approaches is using pattern detection that decreases the system overhead and execution time for malware prediction. The main disadvantage of the signature-based detection approaches is omitting feature selection. The target environment is categorized into three main platforms including embedded systems, Windows-based and smartphones. The most research studies in the signature-based detection have used the Windows-based environment for representing the proposed malware detection approach.

In addition, Table 4 depicts a side-by-side comparison of the signature-based detection factors in each article. These factors include case-study method, classification or clustering approach, data analysis method, and data set type and accuracy factor.

Review of the selected behavior-based approaches

Altaher [38] proposed an evolving hybrid neuro-fuzzy classifier (EHNFC) for Android-based malware grouping utilizing consent based components. The proposed EHNFC not just has the capacity of distinguishing obscured malware utilizing fluffy tenets, yet can likewise advance its structure by adopting new malware recognition fluffy tenets to enhance its discovery exactness when utilized as a part of the location of more malware applications. To this end, a developing bunching technique for adjusting and advancing malware location fluffy tenets was changed to consolidate a versatile methodology for overhauling the radii and focuses of grouped authorization based components. This adjustment to the advancing bunching strategy improves group merging also, produces decides that are better custom-made to the input information, henceforth enhancing the characterization precision of the proposed EHNFC. The exploratory outcomes for the proposed EHNFC demonstrate that the proposition outflanks a few cutting-edge jumbled malware order approaches as far as a false negative rate (0.05) and false positive rate (0.05). The outcomes likewise show that the proposition identifies the Android malware superior to other neuro-fuzzy frameworks as far as precision (90%).

Mohaisen et al. [39] proposed, a computerized and conduct based malware examination and marking framework called AMAL that addresses shortcomings of the current frameworks. AMAL comprises of two sub-frameworks, AutoMal and MaLabel. AutoMal gives instruments to gather low granularity behavioral curios that portray malware utilization of the document framework, memory, organize, what's more, registry, and does that by running malware tests in virtualized situations. On the other hand, MaLabel utilizes those ancient rarities to make delegate highlights, utilize them for building classifiers prepared by physically screened preparing tests, and utilize those classifiers to characterize malware tests into families comparable in conduct. AutoMal additionally empowers unsupervised learning, by executing various bunching calculations for tests gathering. An assessment of both AutoMal and MaLabel in view of medium-scale

Table 3 A side-by-side comparison of the reviewed signature-based articles

Method	Main idea	Advantages	Disadvantages	Target environment
PMD	Polymorphic Malware Detection (PMD) [25]	Low cost High accuracy	Increasing total feature selection	Windows-based
SigPID	Significant permission identification android malware detection (SigPID) [19]	Low cost High accuracy	Low scanning	Smartphone
OpCode	Graph malware detection [3]	Low complexity Low cost	Low timely High robustness	Embedded systems
Droid	Droid malware detection [11]	Fast feature selection	High complexity	Smartphone
APMD	API malware detection (APMD) [23]	Low monitoring overhead High accuracy	High cost	Windows-based
SVDD	N-grams malware detection [20]	High detection accuracy	Did not analyzing feature selection	Windows-based
SMD	Smartphone malware detection (SMD) [29]	Combining static malware analysis and dynamic malware analysis Presenting novel the clone and the mutation mechanism	Did not comparing with other classification approaches Low accuracy	Smartphone
SAAM	Symbolic aggregate approximation for malwares (SAAM) [30]	Best packet classification High accuracy Presenting a data transformation method to reduce the space complexity	Did not examine the multiple packing algorithms.	Windows-based
SOMM	Service-Oriented mobile malware detection (SoMM) [31]	High detection accuracy High scaling	High traffic Did not analyzing behavior of malwares	Smartphone
SPM	Sequential pattern mining (SMP) [32]	High accuracy Low overhead	Did not analyzing feature selection	Windows-based
FPM	Frequent pattern mining (FPM) [33]	Presenting automatic train approach	Not analysis discriminative frequent behavior patterns High overhead	Windows-based
MOED	Multi-objective evolutionary detection (MOED) [34]	High speed detection High accuracy Low overhead	Using traditional detection engines	Smartphone
Opcode	Opcode sequences [35]	Prefect detection ratio of unknown malware	Did not analyze instance selection	Smartphone
MobA	Mobile android [24]	Good attribute selection Low overhead	High complexity Did not analysis countermeasures	Smartphone
SHMD	Signature and Heuristic-based malware detection [36]	Low overhead Best binary feature selection	High time complexity High cost	Smartphone
MKLDroid	A multi-view context-aware approach to Android malware detection [15]	High efficiency Run time detection	High complexity Did not analyzing feature selection	Smartphone
DBScan	Hybrid pattern based text mining approach [17]	Low overhead	High time Low scalability	Windows-based
DroidNative	Android malware detector with control flow patterns [37]	Low time High efficiency	Low scalability High cost	Smartphone
BAM	Hybrid malware detection with binary associative memory [13]	High efficiency	High complexity	Windows-based

(4000 specimens) and expansive scale datasets (more than 115,000 samples) collected and broke down via AutoMal shows AMAL's adequacy in precisely describing, ordering, and gathering malware tests. MaLabel accomplishes an exactness of 99.5% and review of 99.6% to confident relations demand, and more than 98% of accuracy and evaluation for unsupervised classification.

Yuan et al. [40] presented a deep learning method to connect the components from the static investigation with elements from the dynamic investigation of Android applications. In addition, they actualized an Android malware detection engine based on the deep-learning method (DroidDetector) that can consequently distinguish whether a file has a malicious behavior or not. With a large number of Android applications, they tested DroidDetector and play out an in-depth examination of the elements that deep learning basically adventures to portray malware completely. The outcomes appear that deep learning is appropriate for characterizing Android malware and particularly compelling with the accessibility of additional preparation information. DroidDetector can accomplish 96.76% detection accuracy, which traditional machine learning methods.

Boukhtouta et al. [41] presented the issue of fingerprinting perniciousness of activity with the end goal of recognition and arrangement. This research pointed first at fingerprinting perniciousness by utilizing two approaches: Deep Packet Inspection (DPI) and IP bundle headers arrangement. To this end, we consider malignant activity created from element malware examination as movement perniciousness ground truth. In light of this supposition, they exhibited how these two methodologies are utilized to recognize what's more, attribute maliciousness to the various threat. In this work, we concentrate the positive and negative angles for Deep Packet Review and IP bundle headers order. They assessed every approach in view of its recognition and attribution precision and additionally their level of multifaceted nature. The results of both methodologies have demonstrated promising outcomes as far as discovery; they are great possibility to constitute a collaboration to expand or prove recognition frameworks as far as runtime speed and grouping exactness.

Ding et al. [42] proposed an affiliation mining strategy based on API calls to recognize malware. To expand the identification speed of the Objective-Oriented association (OOA) mining, distinctive methodologies are exhibited: to enhance the govern quality, criteria for API determination are proposed to expel APIs that can't get to distinctly visit things; to discover affiliation decides that have solid segregation control, we characterize the manage utility to assess the affiliation runs; and to enhance the location exactness, a characterization strategy in view of numerous affiliation guidelines is embraced. The trials demonstrate that the proposed systems can essentially enhance the running velocity of OOA. In our investigations, the time cost for information mining is decreased by 32%, and the time cost for arrangement is decreased by 50%.

Eskandari et al. [43] presented a novel hybrid approach, HDM-Analyzer, is displayed which takes points of interest of dynamic and static investigation techniques for rising pace while protecting the precision at a sensible level. HDM-Analyzer can foresee the dominant part of basic leadership focuses on using the factual data which is assembled by element investigation; along these lines, they have no any performance overhead. The fundamental commitment of this paper is taking exactness preferred standpoint of the element investigation and consolidating it into static examination keeping in mind

Table 4 A side-by-side comparison of the important factors in the signature-based detection of each article

Case study	Classification approach	Data analysis method	Used dataset	Total dataset	Accuracy %
Polymorphic Malware Detection [25]	K-means	Dynamic	ClamAV, VirusTotal,	2876	99
Android malware detection [19]	SVM	Dynamic	Google play store	5494	94
Graph malware detection [3]	Graph-SVM	Dynamic	Windows DLL calls	6671	88
Droid malware detection [11]	SVM	Dynamic	Windows API library	7000	98
API malware detection [23]	Naive Bayes and Decision Tree—SVM	Dynamic	Google play store	7000	95
N-grams malware detection [20]	SVM	Dynamic	Google play store	658	97
Smartphone malware detection [29]	K-means—artificial immune system	Hybrid	Android malware database XVNA	1300	89.8
Symbolic aggregate approximation for malwares [30]	Naive Bayes and SVM	Dynamic	Offensive computing and VX heavens library	8100	95.83
Service-Oriented mobile malware detection [31]	Naive Bayes and Decision Tree	Hybrid	Key Laboratory of Network Security, Fujian Normal University	3000	97.3
Sequential pattern mining [32]	All-Nearest-Neighbor, KNN, SVM J48	Hybrid	VXHeaven website	3200	95.2
Frequent pattern mining [33]	Minimal contrast frequent subgraphs	Static	Several websites	2083	92
Multi-objective evolutionary detection [34]	Multi-objective evolutionary by GA	Static	Viruseshair and VirusTotal websites	9383	95.15
Opcode sequences [35]	K-nearest neighbors and SVM	Hybrid	VxHeavens website	2000	92.9
Mobile android [24]	SVM	Hybrid	Contagio Blogger and VirusTotal Web sites	2500	98.7
Signature and Heuristic-based Malware Detection [36]	SVM, J48, KNN, Decision tree and Random tree	Hybrid	M0DROID website	500	99.81
A multi-view [15] context-aware approach to Android malware detection	Multiple Kernel Learning, SVM	Static	Google Play, AndroidDrawer, FDroid	6056	98.05
Hybrid pattern based text mining approach [17]	ANN, malicious sequential pattern based malware detection	Hybrid	Viruseshair and VirusTotal websites	8000	98.89
Android malware detector with control flow patterns [37]	Droid, CFGO-IL	Static	Several websites	3158	93.57
Hybrid malware detection with binary associative memory [13]	MLP, SVM, Naïve Bayes, J48	Hybrid	VX Heaven website	52,183	98.6

the end goal to enlarge the precision of static investigation. Truth be told, the execution overhead has been endured in learning stage; hence, it does not force on highlight extraction stage which is performed in examining operation. The exploratory outcomes illustrate that HDM-Analyzer accomplishes better general exactness and time many-sided quality than static and element investigation strategies.

Miao et al. [44] presented a bilayer conduct reflection strategy in light of the semantic examination of dynamic API sequences. Operations on touchy framework assets and complex practices are disconnected in an interpretable way at various semantic layers. At the lower layer, crude API calls are joined to extract low-layer practices by means of information reliance investigation. At the higher layer, low-layer practices are further joined to build more intricate high-layer practices with great interpretability. The separated low-layer furthermore, high-layer practices are at last inserted into a high dimensional vector space. Henceforth, the disconnected practices can be specifically utilized by numerous prominent machine learning calculations. In addition, to handle the issue that considerate projects are not satisfactorily examined or malware and amiable projects are seriously imbalanced, an enhanced one-class bolster vector machine (OC-SVM) named OC-SVM-Neg is proposed which makes utilization of the accessible negative examples. The trial comes about demonstrate that the proposed include extraction technique with OC-SVM-Neg beats double classifiers on the false caution rate and the speculation capacity.

Ming et al. [45] have presented a substitution attacks to cover comparable practices by harming behavior-based specifications. The key strategy for the attacks is to supplant a system call dependence graph to its semantically identical variations so that the comparable malware tests confidential unique family end up being characteristic. Accordingly, malware investigators need to put more endeavors into reconsidering the similar samples which may have been examined sometime recently. They distill general attacking strategies by mining more than 5200 malware tests' behavior specifications and execute a compiler-level model to automate replacement attacks. By evaluating on the real malicious examples, the effectiveness of the proposed method to obstruct several behavior-based malware analysis tasks, such as clustering and malware comparison. Finally, they discussed likely countermeasures to support current malware protection.

Nikolopoulos and Polenakis [46] have proposed a graph-based model which using relations between gatherings of system-calls, distinguishes whether an unknown software sample is malicious or benign, and classifies a malevolent software to one of a set of an arrangement of known malware families. All the more correctly, clients used the System-call Dependency Graphs (or, for short, ScD-graphs), acquired by traces captured through dynamic taint investigation. The authors planed their model to be safe against strong changes applying our recognition and arrangement systems on a weighted coordinated graph, to be specific Group Relation Graph, or Gr-graph for short, coming about because of ScD-graph subsequent to gathering disjoint subsets of its vertices. For the discovery procedure, the authors proposed the Delta-comparability metric, and for the procedure of classification, they proposed the SaMe-similitude and NP-similarity measurements comprising the SaMe-NP closeness. At last, they evaluated their model for malware recognition and classification demonstrating its possibilities against malicious software measuring its identification rates and classification accuracy.

Sheen et al. [47] have considered Android-based malware for examination and an adaptable recognition component is planned to utilize multi-feature collaborative decision fusion (MCDF). The distinctive features of a malicious record like the consent-based features and the API call based features are considered keeping in mind the end goal to give a superior discovery via preparing a gathering of classifiers and combining their choices utilizing collective approach in view of likelihood hypothesis. The execution of the proposed model is evaluated on a gathering of Android-based malware including diverse malware families and the outcomes demonstrate that the presented approach give a superior execution than best in class troupe plans accessible.

Norouzi et al. [48] have proposed distinctive classification techniques with a specific end goal to recognize malware in light of the element and conduct of each malware. A dynamic investigation technique has been exhibited for recognizing the malware features. A recommended program has been introduced for changing over a malware behavior executive history XML document to an appropriate WEKA instrument input. To represent the execution proficiency and preparing information and test, the authors apply the proposed ways to deal with a genuine contextual investigation information set utilizing WEKA instrument. The evaluation results described that the availability of the proposed data mining approach. In addition, their proposed data mining methodology is more proficient for identifying malware and behavioral classification of malware can be helpful to recognize malware in a behavioral antivirus.

Galal et al. [49] proposed a behavior-based features model that defines malicious action exhibited by malware example. To remove the proposed model, the authors first perform dynamic examination on a generally late malware dataset inside a controlled virtual environment and capture traces of API calls conjured by malware examples. The traces are then generalized into high-level features refer to as actions. The proposed method is evaluated using some famous classification methods such as random forests, decision tree and SVM. The experimental results show that the classifiers attain high precision and satisfactory results in the detection of malware variants.

Summary of the reviewed behavior-based approaches

According to the discussed and reviewed behavior-based detection approaches, the comparison of the proposed articles has illustrated in Table 5. Table 5 presents the main idea, advantages, disadvantages and target environment of each technical study in behavior-based approaches. The main advantage of behavior-based detection approaches is detecting all of the suspicious files according to their calls' behavior that increases the accuracy of malware prediction. The main disadvantage of the signature-based detection approaches is the runtime overhead. The target environment is categorized into three main platform including embedded systems, windows-based and smartphones. The most research studies in the behavior-based detection have used the smartphone environment for representing the proposed malware detection approach.

Also, Table 6 shows a technical comparison of the behavior-based detection factors in each article. These factors include case-study method, classification or clustering approach, data analysis method, used data set, total number of dataset and accuracy factor.

Table 5 A comparison of the reviewed behavior-based articles

Method	Main idea	Advantages	Disadvantages	Target environment
DeepAM	Deep learning malware detection [9]	Solving the encrypted Problem in malware detection Higher accuracy	High cost High timely	Embedded systems
QDFG	Graph mining in malware detection [21]	Reducing response time	High complexity High cost	Smartphone
DMDAM	Android malware detection [6]	Reducing concepts for increasing feature selection High accuracy	High complexity Run-time overhead	Smartphone
AMP	Android malware detection [22]	High accuracy	High cost	Smartphone
AMD	Android malware detection [38]	Higher accuracy than the other neuro-fuzzy approaches Minimum false positive and false negative	Did not considering dynamic analysis of Android apps Run-time overhead	Smartphone
AMAL	AMAL: automated malware analysis [39]	Providing high levels of precision, recall, and accuracy Low cost	IP reputation High overhead	Smartphone
AMCS	Android Malware Characterization and Detection [40]	Conducting static and dynamic analyses to extract features from each applications Deploying online testing for Droid-detector	High cost High overhead on API calls	Smartphone
DPIM	Deep Packet Inspection for malware [41]	High classification accuracy Independence from packet payloads Decoupling between detection and attribution	Datasets over fitting High complexity	Windows-based
OOM	Objective Oriented malware [42]	Adapting multiple association rules Improve the running speed of classification	High complexity High cost Not analyzing unmatched files	Windows-based
HAM	Hybrid analysis malware [43]	Low execution overhead High accuracy time	High time consumption	Windows-based
BBA	Bilayer behavior abstraction [44]	Low overhead	Did not analyzing feature selection	Windows-based
Mspec	Malware specifications [45]	Good normalizing features Low execution time	Did not analyzing the accuracy conditions High complexity	Windows-based
SyCM	System-call malware [46]	High accuracy High dependency analysis for calls	High time consumption	Smartphone
ABM	Android based malware [47]	Using multi-feature attributes High scalability	High complexity High execution time	Smartphone
DBM	Behavioral malware [48]	Extracting XML to feature files High scalability	High complexity	Windows-based
MAPI	Malicious code based on API [49]	Adding additional heuristic occupations to show more actions High accuracy rates	Not suitable for samples of external events Existence analysis	Windows-based

Table 5 continued

Method	Main idea	Advantages	Disadvantages	Target environment
CloudIntell	Feature extraction method in cloud [18]	Lowest energy consumption High scalability	High complexity High response time	Windows-based
SDMS	Security dependency network for malware detection [50]	Low response time High accuracy	High energy High complexity	Windows-based
DFAMD	Data flow android malware detection [51]	High efficiency Low overhead Low time	High complexity High dependency	Smartphone
SCCMD	So-called compression-based malware detection [21]	High efficiency Low complexity	High response time	Windows-based
DeepFlow	Deep-learning malware detection [52]			Smartphone

Discussion

In this section, a statistical analysis of reviewed approaches of malware detection using data mining is presented. Figure 4 shows the statistical diagram for all of the classification methods in the selected malware detection approaches. In this report, the SVM method has most percentage for malware detection approach with 29%, j48 has 17%, NB has 10%, RF has 5%, ANN has 3% and the other methods have less than 2% usage in data mining results. We discover that the SMV method just has the best accuracy in the signature-based malware detection approaches using data mining.

Also, Fig. 5 shows the accuracy factor for each research. As shown, all of the accuracy factors higher than 80%. The maximum accuracy percentage is 99.2% for the DPIM approach [41] and the minimum accuracy percentage is 86% for the DMDAM approach [22].

Also, Fig. 6 shows the main case study diagram of each research in malware detection. As shown, the recent researches have considered android smartphones to analyze malware detection approaches with 40%. The symbolic code aggregation case studies in windows-based platform has 23%, the pattern mining has 11%, the system calls has 8% usage in malware detection.

In addition, Fig. 7 illustrates the total number of data set used for malware detection analysis in each research. In this figure, there are five research that use higher than 5000 real samples during the evaluation process. The BBA approach [44] has the maximum dataset with 17,000 samples and the AMD approach [38] has the minimum dataset with 500 samples.

Also, Fig. 8 shows the data analysis methods percentage in terms of static, dynamic and hybrid analysis in selected research. The most data analysis methods have used dynamic analysis with 51%, the hybrid analysis has 29% and the static analysis has 20% usage. The 30% of the signature-based approaches have used the dynamic data analysis. The 65% of the behavior-based malware detection approaches have used the dynamic data analysis method.

Table 6 A side-by-side comparison of the important factors in behavior-based detection of each article

Case study	Classification approach	Data analysis method	Used dataset	Total dataset	Accuracy %
Deep learning malware detection [9]	DeepAM	Dynamic	Windows API calls in Comodo Cloud Security Center	2000	98
Graph mining in malware detection [21]	Graph search	Dynamic	Windows sandbox malware	6994	96
Android malware detection [6]	Random forest	Dynamic	Android applications	170	86
Android malware detection [22]	Multilayer perceptron	Dynamic	Several websites	734	97
Android malware detection [38]	Evolving neuro-fuzzy inference system	Dynamic	Google play and android Malware genome Project	500	90
AMAL: automated malware analysis [39]	Decision trees	Dynamic	Random sample from internal user and external customers such as antivirus companies	2086	98
Android malware characterization and detection [40]	Deep belief networks	Hybrid	Google play and android Malware genome project	1860	96.76
Deep Packet Inspection for malware [41]	BoostedJ48, J48, Naïve Bayesian and SVM	Dynamic	Wireless and Secure Networks Research Lab	4560	99
Objective Oriented malware [42]	Multiple association rules	Hybrid	Several websites	8000	97.2
Hybrid analysis malware [43]	Bayesian network, Naive Bayes, Lazy K-Stare	Hybrid	Selected randomly from malware repository of APA, the security research laboratory at Shiraz University	3000	95.27
Bilayer behavior abstraction [44]	SMV, Naïve Bayes, decision tree, logistic regression	Dynamic	Open-access malware database such as VXHeaven website	17,000	94
Malware specifications [45]	System call dependency graph	Dynamic	VXHeavens website	5200	92
System-call malware [46]	SaMe-NP	Dynamic	Variety of commodity software types including editors, office suites, media players,	2667	95.9
Android based malware [47]	J48, SVM, IBk, NaiveBayes	Static	Google play and android Malware services	2000	98.91
Behavioral Malware [48]	Regression, SVM, J48	Dynamic	Web data commons library in VirusSign and VXHeaven	7000	98.3

Table 6 continued

Case study	Classification approach	Data analysis method	Used dataset	Total dataset	Accuracy %
Malicious code based on API [49]	Decision tree, SVM and random forest	Dynamic	API hooking library in VirusSign	2000	96.89
Feature extraction method in cloud [18]	Decision tree, SVM, Boosting	Static	Random dataset of VirusTotal	15,000	99.69
Security dependency network for malware detection [50]	No read down and no write up	Dynamic	VXHeavens website	7257	93.92
Data flow android malware detection [51]	KNN, LR, BN	Static	VXHeavens website and Google play	2200	97.66
So-called compression-based malware detection [21]	k-NN, QDA, LDA, SVN, Decision Trees, and random forest	Dynamic	Cuckoo sandbox	7507	99.3
Deep-learning malware detection [52]	Naive Bayes, PART, Logistic Regression, SVM and MLP	Hybrid	Google play, virus share	11,000	95.05

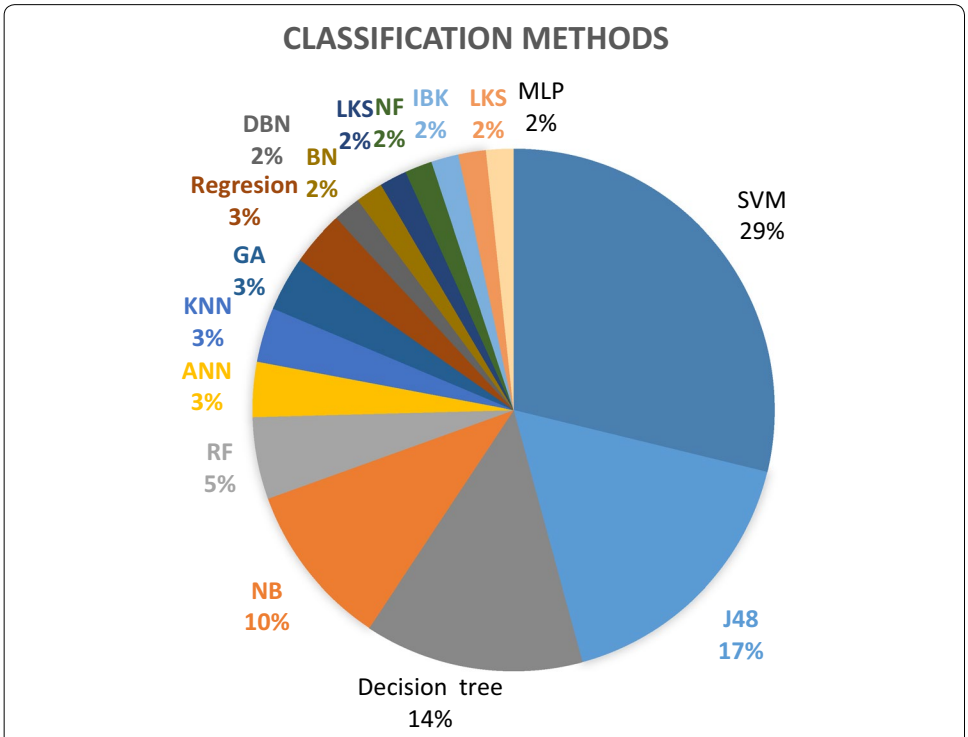

Fig. 4 Classification methods in malware detection mechanism

Open issues

Due to applying the survey on the malware detection approaches, the following research challenges as the open issues are presented that are not addressed by the research populations up to now.

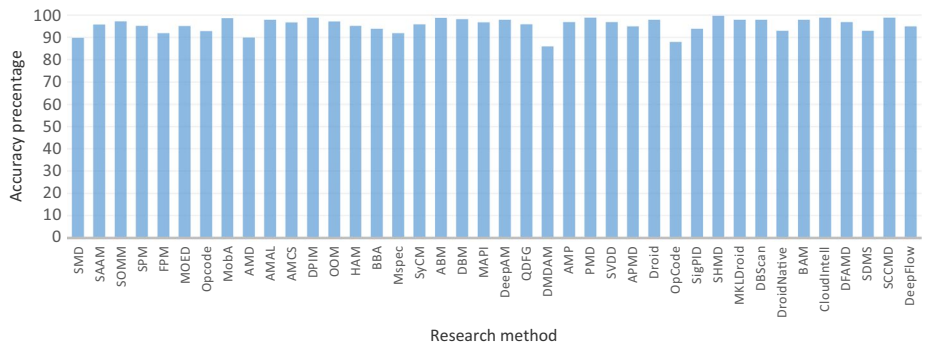

Fig. 5 Accuracy factor for selected approaches in malware detection

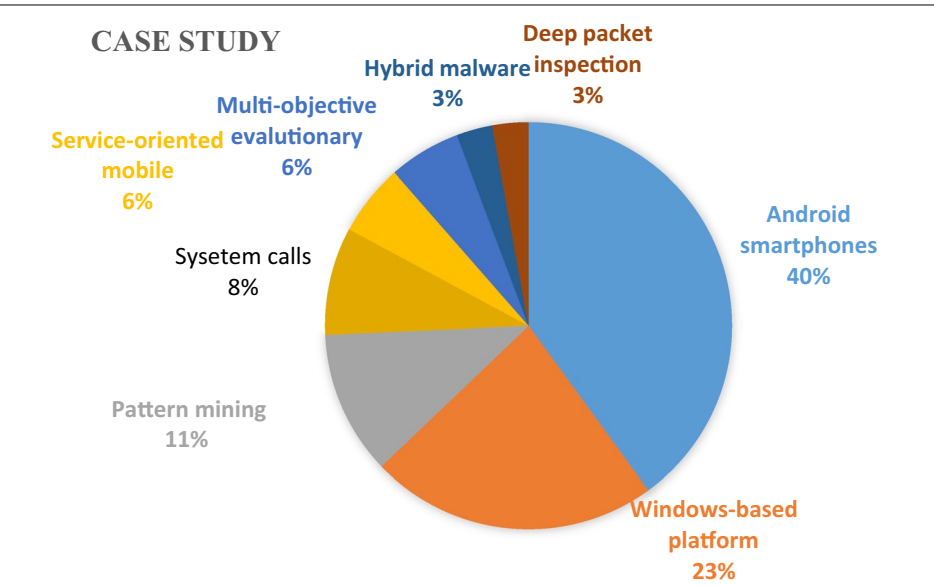

Fig. 6 Case study analysis for each research in malware detection

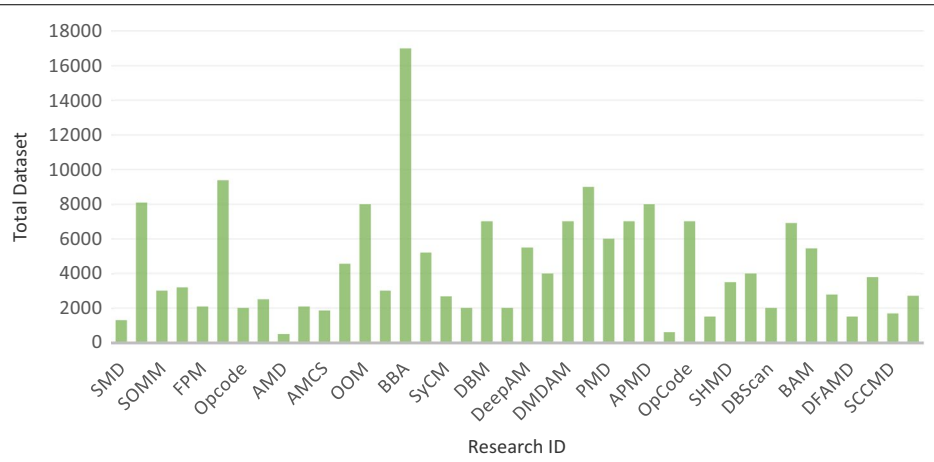

Fig. 7 The total number of dataset used in each research

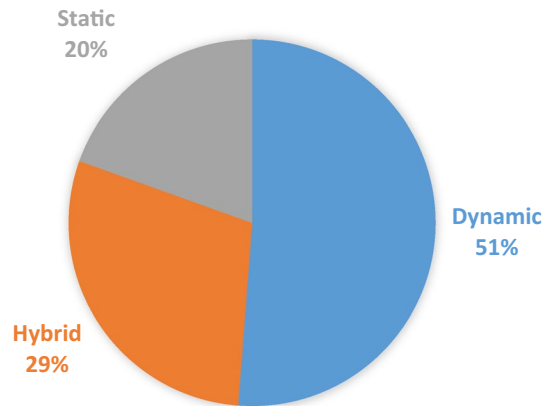

Fig. 8 The data analysis methods in the selected articles

- Decryption/encryption detection: One of the important open issues in malware detection is information hiding malware techniques. Information hiding techniques are utilized to make information hard to take note. This practice ought not to be mistaken for encryption, in which the substance is disjointed, as it is rather clear. Such components are regularly utilized mutually to guarantee that a discussion stays indiscernible. Steganography is a standout amongst the most surely understood subfields of data stowing away and means to shroud mystery information in an appropriate transporter.
- Meta-heuristic detection: The malware detection analyses using meta-heuristic algorithms can influence the speed up of the execution time and the total accuracy factor of the data mining process.
- Real-time malware detection: Is based on hybrid analysis, secure multi-objective evolutionary malware detection, secure e-banking environments and secure healthcare systems are very challenging to recognize the malicious files and hidden attacks using data mining approaches.

Further studies are suggested to improve the accuracy of the related malware detection methods using evolutionary mechanisms.

In this survey, we performed a full description research to find more than 35 authors and different works. However, by considering the increasing development of studies on this topic, it is not possible to guarantee that all of the articles were recovered, particularly for 2010, because the research finished in July 2017.

Suggestion criteria

According to the existing discussion analysis, some technical suggestions are introduced to expand the malware detection approaches in the new platforms and architectures such as Internet of Things (IoT) applications, e-banking and social networks.

Some evolutionary methods can be improve the malware detection for predicting the polymorphism attacks in the electronic wallet applications. For example, a meta-heuristic algorithm finds the optimal signature detection for a polymorphism malware attacks in the electronic mobile payments.

Context-aware detection is a new idea for dynamic malware detection approaches in the IoT applications based on semantic signature that categorize API calls with respect to the most interactions between end user and application layer of the IoT. When the smart devices cannot interact between user devices and datacenters, the reliability and availability of the smart services have been decreased.

Providing a safe condition for the huge data collection such as big data against the malware attacks is the key challenge for the malware detection for navigating big data security. Therefore, to select the minimal sample space of the malware damage, the data collection and storing big data can be navigate using data mining and synthesis methods.

Conclusion

This paper presented a systematic literature survey of the malware detection approaches using data mining. The reviewed and papers were investigated and classified into two main categories; (1) signature-based and (2) behavior-based approaches. The malware detection approaches were compared and analyzed according to various essential factors such as classification approaches, data analysis methods, the number of the used dataset, accuracy factor and case study analysis. The advantage and disadvantage of each method were deliberated in the malware detection methods. Most of the selected articles in data mining are behavior-based techniques. In the malware analysis stage, the most case studies are proposed for the android smartphones. In addition, using metaheuristic algorithms in malware detection analysis can speed up and improve the execution time and the overall accuracy of the data mining process. As the experimental results, we observed that the SVM method has most percentage for malware detection approach with 29%, j48 has 17%, Decision tree has 14%, NB has 10%, BF has 5% and the other methods have less than 3% usage in data mining results. We discover that the SVM method just has the best accuracy in the signature-based malware detection approaches using data mining. In addition, the maximum accuracy percentage is 99.2% for the DPIM approach and the minimum accuracy percentage is 86% for the DMDAM approach. Also, we observed that the recent researches have considered android smartphones to analyze malware detection approaches with 40%. The symbolic code aggregation case studies in windows-based platform has 23%, the pattern mining has 11%, the system calls has 8% usage in malware detection. Finally, we have seen that The 30% of the signature-based approaches have used the dynamic data analysis. The 65% of the behavior-based malware detection approaches have used the dynamic data analysis method. As an important open issue, some important topics such as secure multi-objective malware, e-banking environments, and healthcare systems malware attacks are challenging areas to recognize the malicious files and hidden attacks.

Authors' contributions
AS as the corresponding author. RH as the co-author. Both authors read and approved the final manuscript.

Author details
[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. [2] Department of Computer Engineering, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran.

Acknowledgements
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Availability of data and materials
Not applicable.

Consent for publication
Not applicable.

Ethics approval and consent to participate
We confirm that this manuscript has not been published elsewhere and is not under consideration by another journal. All authors have approved the manuscript and agree with its submission.

Funding
Not applicable.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 20 July 2017 Accepted: 2 January 2018
Published online: 12 January 2018

References

1. Souri A, Norouzi M, Asghari P (2017) An analytical automated refinement approach for structural modeling large-scale codes using reverse engineering. Int J Inf Technol 9:329–333. https://doi.org/10.1007/s41870-017-0050-7
2. Souri A, Navimipour NJ, Rahmani AM (2017) Formal verification approaches and standards in the cloud computing: a comprehensive and systematic review. Comput Stand Interfaces. https://doi.org/10.1016/j.csi.2017.11.007
3. Hashemi H, Azmoodeh A, Hamzeh A, Hashemi S (2017) Graph embedding as a new approach for unknown malware detection. J Comput Virol Hacking Tech 13:153–166. https://doi.org/10.1007/s11416-016-0278-y
4. Park JH (2017) Novel approaches for applying linguistic processing techniques based on pattern recognition and machine learning. JIPS (J Inf Process Syst) 13:643–652
5. Souri A, Asghari P, Rezaei R (2017) Software as a service based CRM providers in the cloud computing: challenges and technical issues. J Serv Sci Res 9:219–237. https://doi.org/10.1007/s12927-017-0011-5
6. Bhattacharya A, Goswami RT (2017) DMDAM: data mining based detection of android malware. In: Mandal JK, Satapathy SC, Sanyal MK, Bhateja V (eds) Proceedings of the first international conference on intelligent computing and communication springer Singapore, Singapore, pp 187–194
7. Nikolopoulos SD, Polenakis I (2017) A graph-based model for malware detection and classification using system-call groups. J Comput Virol Hacking Tech 13:29–46. https://doi.org/10.1007/s11416-016-0267-1
8. Pektaş A, Acarman T (2017) Classification of malware families based on runtime behaviors. J Inf Secur Appl 37:91–100. https://doi.org/10.1016/j.jisa.2017.10.005
9. Ye Y, Chen L, Hou S, Hardy W, Li X (2017) DeepAM: a heterogeneous deep learning framework for intelligent malware detection. Knowl Inf Syst. https://doi.org/10.1007/s10115-017-1058-9
10. Safarkhanlou A, Souri A, Norouzi M, Sardroud SEH (2015) Formalizing and verification of an antivirus protection service using model checking. Procedia Comput Sci 57:1324–1331. https://doi.org/10.1016/j.procs.2015.07.443
11. Li Z, Sun L, Yan Q, Srisa-an W, Chen Z (2017) DroidClassifier: efficient adaptive mining of application-layer header for classifying android malware. In: Deng R, Weng J, Ren K, Yegneswaran V (eds) Security and privacy in communication networks: 12th international conference, securecomm 2016, Guangzhou, China, October 10–12, 2016, Proceedings. Springer International Publishing, Cham, pp 597–616
12. Malhotra R, Jangra R (2017) Prediction & assessment of change prone classes using statistical & machine learning techniques. J Inf Process Syst 13(4):778–804. https://doi.org/10.3745/JIPS.04.0013
13. Chowdhury M, Rahman A, Islam R (2018) Malware analysis and detection using data mining and machine learning classification. In: Abawajy J, Choo K-KR, Islam R (eds) International conference on applications and techniques in cyber security and intelligence: applications and techniques in cyber security and intelligence. Springer International Publishing, Cham, pp 266–274
14. Palumbo P, Sayfullina L, Komashinskiy D, Eirola E, Karhunen J (2017) A pragmatic android malware detection procedure. Comput Secur 70:689–701. https://doi.org/10.1016/j.cose.2017.07.013
15. Narayanan A, Chandramohan M, Chen L, Liu Y (2017) A multi-view context-aware approach to Android malware detection and malicious code localization. Empir Softw Eng. https://doi.org/10.1007/s10664-017-9539-8
16. Mohamed GAN, Ithnin NB (2018) SBRT: API signature behaviour based representation technique for improving metamorphic malware detection. In: Saeed F, Gazem N, Patnaik S, Saed Balaid AS, Mohammed F (eds) Recent trends in information and communication technology. Proceedings of the 2nd international conference of reliable information and communication technology (IRICT 2017). Springer International Publishing, Cham, pp 767–777
17. Malhotra A, Bajaj K (2016) A hybrid pattern based text mining approach for malware detection using DBScan. CSI Trans ICT 4:141–149. https://doi.org/10.1007/s40012-016-0095-y
18. Siddiqui M, Wang MC, Lee J (2008) A survey of data mining techniques for malware detection using file features. In: Proceedings of the 46th annual southeast regional conference on xx. 2008. ACM
19. Sun L, Li Z, Yan Q, Srisa-an W, Pan Y (2016) SigPID: significant permission identification for android malware detection. In: 2016 11th international conference on malicious and unwanted software (MALWARE), pp 1–8
20. Boujnouni ME, Jedra M, Zahid N (2015) New malware detection framework based on N-grams and support vector domain description. In: 2015 11th international conference on information assurance and security (IAS), pp 123–128

21. Wuechner T, Cislak A, Ochoa M, Pretschner A (2017) Leveraging compression-based graph mining for behavior-based malware detection. IEEE Trans Dependable Secur Comput. https://doi.org/10.1109/tdsc.2017.2675881
22. Bhattacharya A, Goswami RT (2017) Comparative analysis of different feature ranking techniques in data mining-based android malware detection. In: Satapathy SC, Bhateja V, Udgata SK, Pattnaik PK (eds) Proceedings of the 5th international conference on frontiers in intelligent computing: theory and applications: FICTA 2016, Volume 1. Springer Singapore, Singapore, pp 39–49
23. Fan CI, Hsiao HW, Chou CH, Tseng YF (2015) Malware detection systems based on API log data mining. In: 2015 IEEE 39th annual computer software and applications conference, pp 255–260
24. Wang P, Wang Y-S (2015) Malware behavioural detection and vaccine development by using a support vector model classifier. J Comput Syst Sci 81:1012–1026. https://doi.org/10.1016/j.jcss.2014.12.014
25. Fraley JB, Figueroa M (2016) Polymorphic malware detection using topological feature extraction with data mining. In: SoutheastCon 2016, pp 1–7
26. Sun M, Li X, Lui JC, Ma RT, Liang Z (2017) Monet: a user-oriented behavior-based malware variants detection system for android. IEEE Trans Inf Forensics Secur 12:1103–1112
27. Sun H, Wang X, Buyya R, Su J (2017) CloudEyes: cloud-based malware detection with reversible sketch for resource-constrained internet of things (IoT) devices. Softw Pract Exp 47:421–441. https://doi.org/10.1002/spe.2420
28. Tang Y, Xiao B, Lu X (2011) Signature tree generation for polymorphic worms. IEEE Trans Comput 60:565–579. https://doi.org/10.1109/TC.2010.130
29. Wu B, Lu T, Zheng K, Zhang D, Lin X (2014) Smartphone malware detection model based on artificial immune system. China Commun 11:86–92. https://doi.org/10.1109/CC.2014.7022530
30. Bat-Erdene M, Park H, Li H, Lee H, Choi MS (2017) Entropy analysis to classify unknown packing algorithms for malware detection. Int J Inf Secur 16(3):227–248. https://doi.org/10.1007/s10207-016-0330-4
31. Cui B, Jin H, Carullo G, Liu Z (2015) Service-oriented mobile malware detection system based on mining strategies. Pervasive Mob Comput 24:101–116. https://doi.org/10.1016/j.pmcj.2015.06.006
32. Fan Y, Ye Y, Chen L (2016) Malicious sequential pattern mining for automatic malware detection. Expert Syst Appl 52:16–25. https://doi.org/10.1016/j.eswa.2016.01.002
33. Hellal A, Romdhane LB (2016) Minimal contrast frequent pattern mining for malware detection. Comput Secur 62:19–32. https://doi.org/10.1016/j.cose.2016.06.004
34. Martín A, Menéndez HD, Camacho D (2016) MOCDroid: multi-objective evolutionary classifier for Android malware detection. Soft Comput 21:7405–7415. https://doi.org/10.1007/s00500-016-2283-y
35. Santos I, Brezo F, Ugarte-Pedrero X, Bringas PG (2013) Opcode sequences as representation of executables for data-mining-based unknown malware detection. Inf Sci 231:64–82. https://doi.org/10.1016/j.ins.2011.08.020
36. Rehman Z-U, Khan SN, Muhammad K, Lee JW, Lv Z, Baik SW, Shah PA, Awan K, Mehmood I (2017) Machine learning-assisted signature and heuristic-based detection of malwares in Android devices. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2017.11.028
37. Alam S, Qu Z, Riley R, Chen Y, Rastogi V (2017) DroidNative: automating and optimizing detection of Android native code malware variants. Comput Secur 65:230–246. https://doi.org/10.1016/j.cose.2016.11.011
38. Altaher A (2016) An improved Android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (EHNFC) and permission-based features. Neural Comput Appl 28:4147–4157. https://doi.org/10.1007/s00521-016-2708-7
39. Mohaisen A, Alrawi O, Mohaisen M (2015) AMAL: high-fidelity, behavior-based automated malware analysis and classification. Comput Secur 52:251–266. https://doi.org/10.1016/j.cose.2015.04.001
40. Yuan Z, Lu Y, Xue Y (2016) Droiddetector: android malware characterization and detection using deep learning. Tsinghua Sci Technol 21:114–123. https://doi.org/10.1109/TST.2016.7399288
41. Boukhtouta A, Mokhov SA, Lakhdari N-E, Debbabi M, Paquet J (2016) Network malware classification comparison using DPI and flow packet headers. J Comput Virol Hacking Tech 12:69–100. https://doi.org/10.1007/s11416-015-0247-x
42. Ding Y, Yuan X, Tang K, Xiao X, Zhang Y (2013) A fast malware detection algorithm based on objective-oriented association mining. Comput Secur 39(Part B):315–324. https://doi.org/10.1016/j.cose.2013.08.008
43. Eskandari M, Khorshidpour Z, Hashemi S (2013) HDM-Analyser: a hybrid analysis approach based on data mining techniques for malware detection. J Comput Virol Hacking Tech 9:77–93. https://doi.org/10.1007/s11416-013-0181-8
44. Miao Q, Liu J, Cao Y, Song J (2016) Malware detection using bilayer behavior abstraction and improved one-class support vector machines. Int J Inf Secur 15:361–379. https://doi.org/10.1007/s10207-015-0297-6
45. Ming J, Xin Z, Lan P, Wu D, Liu P, Mao B (2016) Impeding behavior-based malware analysis via replacement attacks to malware specifications. J Comput Virol Hacking Tech 13:193–207. https://doi.org/10.1007/s11416-016-0281-3
46. Nikolopoulos SD, Polenakis I (2016) A graph-based model for malware detection and classification using system-call groups. J Comput Virol Hacking Tech 13:29–46. https://doi.org/10.1007/s11416-016-0267-1
47. Sheen S, Anitha R, Natarajan V (2015) Android based malware detection using a multifeature collaborative decision fusion approach. Neurocomputing 151(Part 2):905–912. https://doi.org/10.1016/j.neucom.2014.10.004
48. Norouzi M, Souri A, Samad Zamini M (2016) A data mining classification approach for behavioral malware detection. J Comput Netw Commun 2016:9. https://doi.org/10.1155/2016/8069672
49. Galal HS, Mahdy YB, Atiea MA (2016) Behavior-based features model for malware detection. J Comput Virol Hacking Tech 12:59–67. https://doi.org/10.1007/s11416-015-0244-0
50. Mao W, Cai Z, Towsley D, Feng Q, Guan X (2017) Security importance assessment for system objects and malware detection. Comput Secur 68:47–68. https://doi.org/10.1016/j.cose.2017.02.009
51. Wu S, Wang P, Li X, Zhang Y (2016) Effective detection of android malware based on the usage of data flow APIs and machine learning. Inf Softw Technol 75:17–25. https://doi.org/10.1016/j.infsof.2016.03.004
52. Dali Z, Hao J, Ying Y, Wu D, Weiyi C (2017) DeepFlow: deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In: 2017 IEEE symposium on computers and communications (ISCC), pp 438–443