Malware Discernment Using Machine Learning

Vivek Srivastava and Rohit Sharma

1 Introduction

MALWARE can be characterized as the code developed without the permission of the owner to infiltrate or injure an electronic machine [1]. For all kinds of viruses, malware is basically a non-classified description. Computer file contagiousness and autonomous malware constitute a straightforward categorization of malware. Malware can actually also be defined by its particular type: backdoors, worms, spyware, Trojans, rootkits, adware, and so on. Malware detection [5] by machine learning is difficult because all current malware programs seem to have several layers to escape suspicion or use side strategies to simply turn to a newer version at short periods of time in order to prevent them from being recognized by any antivirus subroutines [2].

2 Malware Definition

Viruses, ransomware, and spyware, including a spectrum of different malicious codes, sound like malware as a blanket term. Tachygraphy typically includes code designed by swindlers for malicious applications, built to cause unnecessary system and data interference or to govern unauthorized network access [3]. Generally, malware is transmitted via email in the form of a reference or file that requires the user to tap on the attachment or open the malware executing file.

V. Srivastava (✉) · R. Sharma
Dr. Ambedkar Institute of Technology for Handicapped,
Kanpur, Uttar Pradesh, India

F. Al-Turjman et al. (eds.), *Transforming Management with AI, Big-Data, and IoT*, https://doi.org/10.1007/978-3-030-86749-2_12

3 Latest Trends and Attacks

Here, we present the latest trends and attacks with the reference of an antivirus company Quickheal [6]; in their report, we found the current trends of attacks, some of which we are mentioning in this chapter.

3.1 Ransomware Exploring New Techniques for Process Code Injection

Process code injection is a very popular technique among malware authors to evade security products. Process hollowing is an injection technique where the legitimate process is created in suspended mode, its memory is overwritten with malicious code, and the process is resumed. It seems like a legitimate process performs all the malicious activities, so it is untouched by security products. The new ransomware Mailto or Netwalker is using this old trick in a new way. Instead of creating a process in suspended mode, 'Debug Mode' is used. Using the debug API WaitForDebugEvent, it receives the method and thread information. A segment is then developed with a size that is replicated from the specimen and whole file data. It then manually resolves the relocation. The sample contains an encrypted JSON file in the resource section having required information like a key for generating ID, that is, extension to be added to encrypted files, base64 encoded ransom note, whitelisted paths, and email ids which are part of the extension. The ID is created using the key maintained in decrypted JSON under the 'mpk' tag, the machine name retrieved, and the hardware profile information about the machine being infected. SHA-256 is determined from such inputs, and the first five characters of the result are used as the file extension ID. The ransom note file name is also preserved in the same way as the produced ID. Since the 1980s, malware has posed a threat to servers, networks, and infrastructure. While there are two main technologies to combat this, most organisations depend almost entirely on one technique, the decade-old signature-based technique. The more sophisticated method of detecting malware through behaviour analysis proposed and introduced by the authors in [18] is gaining popularity quickly, but it is still relatively unknown.

3.2 Info-Stealer Hidden in the Phishing Emails!

Cybercriminals [22] pry on the data precious to you! The data consists of your system details, name, software installed, browsing history; cookies stored on the disk; and saved passwords. We have observed multiple phishing emails in this quarter with contents which entice the end user to download a malware encapsulated as fake software or a fake update. These software names have strings like demo, free, cracked or plugin, etc. These malware payloads are often either placed on compromised websites or popular file-sharing service platforms. Once the malware is executed, it starts getting computer details using Windows APIs and stores it in a file. Sometimes, malware downloads a few supporting files, which might be useful for retrieving data – we saw a recent case wherein five to six supporting DLLs were downloaded for Mozilla's Firefox browser. For each kind of data, a separate file is maintained with almost all data stored in a SQLite format. All the stolen data is compressed in a single file and sent to the attacker. We have found some variations in the way of sending data. Some malware contained pre-existing CnC details, and in some cases, there were few mail IDs seen where this data was being sent using Microsoft's CDO library over port 465. To remove the traces, malware finally deletes all the stolen data and downloaded DLLs from the victim's system.

4 Types of Malware

4.1 Virus

Viruses tag their malicious code for cleaning code, the most severe malware method, and look for an inexperienced device user or motorized process to execute the corrupt file. They can expand quickly and widely, like a living virus, posing a danger to the elementary features of systems, manipulating files and preventing authentic users from accessing their computers. They are located within a normal executable set of code.

4.2 Worms

Analogously, they infect computer systems like a real worm plaguing the human body. They amalgamate their way throughout the network, connecting to sequential devices starting from one infected computer to begin the transmission of infection. This kind of malware can swiftly infect common communications systems.

4.3 Spyware

It is intended at spying on what an individual is doing. This malicious code, concealed on a computer in the background, steals processed data without the user's knowledge, such as credentials for bank cards, watchwords and other confidential information.

4.4 Trojans

As licit software, this form of malware stashes inside or dissembles itself. Acting discreetly can cleft protection by building backdoors that offer easy access to other malware variants**.**

4.5 Ransomware

Ransomware is also recognized as spyware and comes at a high cost. Competent of trapping networks and locking out users before a ransom is credited, ransomware has targeted, with exorbitant consequences, some of the giant organizations in the world today.

5 Malware Detection Techniques

Here we have presented the hierarchical structure of malware detection techniques [8, 10] which makes easier to understand the method of malware detection (Fig. 1).

5.1 Signature Based

It is an anti-malware perspective that recognizes the presence of a malware infection or illustration by matching the software in question with at least one-byte code pattern with the set of signature data from established malicious programs, also known as blacklists. This detection scheme is based on the belief that signature-based detection is the most widely used approach for anti-malware systems represented by trends.

5.2 Susceptible to Evasion

These byte patterns are also well documented because the signature byte patterns are fetched from known malware. Therefore, hackers can easily avoid them by using basic garble techniques such as no-ops insertion and re-ordering code. It is, therefore, possible to refit malware code and prevent signature-based detection.

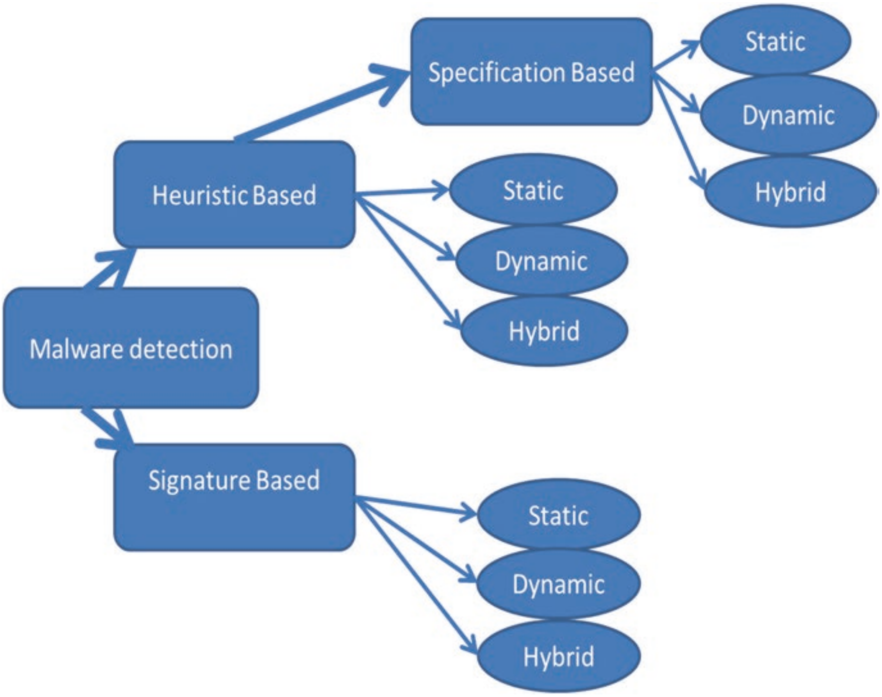


Fig. 1 Malware detection method

5.3 *Zero-Day Attacks*

Zero-Day Attack is an intrusion that targets a potentially significant vulnerability in software security [19] that might not be identified to suppliers or developers. To minimize the vulnerability to the software user, the software developer should fix the vulnerability as soon as possible after it is found. A software improvement is called the alternative. It is also possible to use zero-day attacks to target the Internet of Things.

5.4 *Heuristic Based*

Heuristic evaluation emanates from the primitive Greek word meaning 'to discover', and is a process of exploration, learning and problem-solving that uses semantics, assumptions or informed guesses to find an effective response to a specific issue. Although this problem-solving approach may not be efficient, when applied to computer processes where a fast response or timely warning is required based on inherent prudence, it can be highly efficient. By checking code for mistrustful properties, viruses can be identified through the heuristic analysis method. Reasonable virus detection methods include malware detection by comparing software code to the code of recognized type's pre-identified class of viruses, processed and registered in a set of data known as signature detection. Although useful and still in use, because of the emergence of new threats that have emerged since the turn of the century and remain a problem all the period, the approach of signature detection has also become more limited. The heuristic model was explicitly intended to solve this issue, classify malicious features that can be discovered in unidentified, upcoming viruses and updated versions of existing viruses, and recognized specimens of malware. Progressively, fraudsters are propagating more advanced threats; few methods used to combat the celestial volume of these new threats seen each day. Heuristic analysis is an authentic and powerful tool among these methods.

5.5 Machine Learning Based

Feature Selection [10] is a mechanism used to formulate a hypothesis to pick a proper or appropriate feature. It is primarily used in machine learning, image processing, and identification of patterns, respectively. It is also known as the collection of variables or attributes. When the number of features is very high, this process becomes more important. In this way, we can make our algorithm even better by selecting only the significant and important function. With this, our machine's training and evolution time can be decreased. The collection of features is primarily used for few reasons like machine learning prerequisites can be assembled within the deadline; the Machine Learning Algorithm can be trained to reduce the model's complexity such that the model is easier to understand. Feature selection can be accomplished by various methods like filter method, wrapper method and embedded method (Fig. 2).

Feature selection or attribute selection is very basic and essential part of machine learning in order to develop a model for malware detection because without identifying any feature or proper attribute an authentic model cannot be developed.

6 Malware Analysis Techniques

Malware perusal is the task of determining a deeply suspicious file or URL's actions and intent. The perusal performance assists in identifying and alleviating the significant risk. The main advantage of malware analysis helps event responders and security professionals [11]: Idealistically, triage events by severity level *discover* secret compromise indicators that should be blocked, *increase* the effectiveness of compromise warning and warning indicators and *deepen* sense while hunting for threats (Fig. 3).

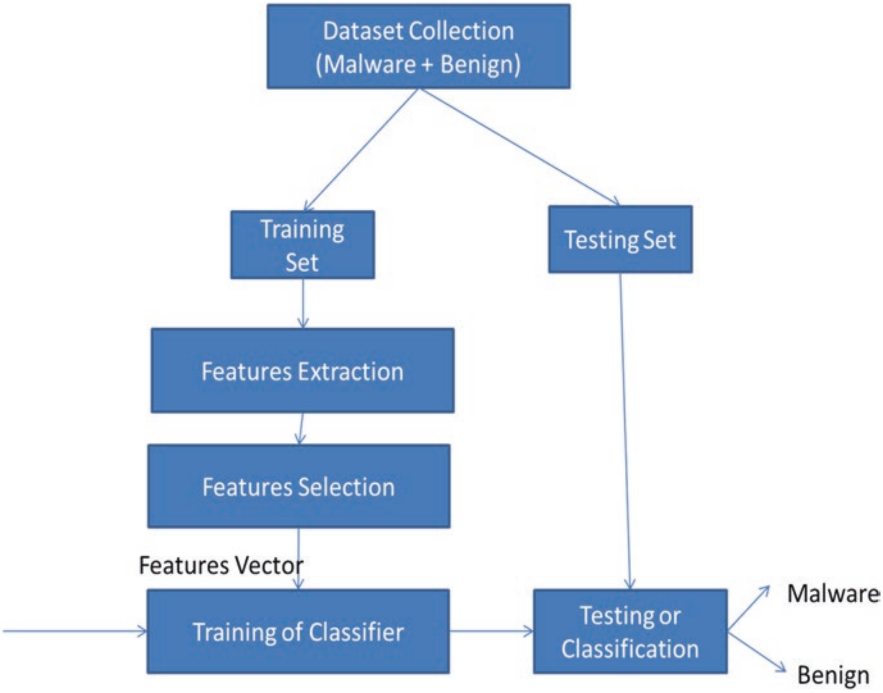
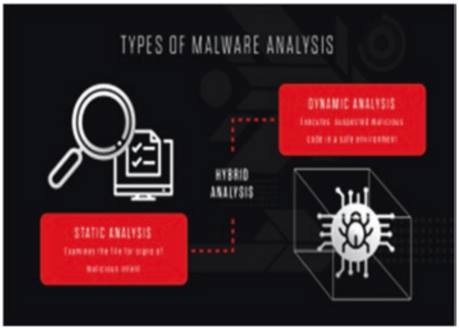


Fig. 2 Machine learning method for identification of malware and benign

Fig. 3 Malware analysis techniques


TYPES OF MALWARE ANALYSIS
DYNAMIC ANALYSIS
Executes suspected malicious code in a safe environment
HYBRID ANALYSIS
STATIC ANALYSIS
Examines the file for signs of malicious intent

6.1 *Static Analysis*

Basic static perusal does not really require the literal implementation of the code [7]. The file is instead investigated by static perusal for signs of malicious intent. Recognizing malicious systems, libraries or packed files may be valuable. This is the procedure through which a binary is analysed without conducting it. It is quick to implement and enables you to retrieve the archive associated with the binary offender. Static perusal may not reveal all the information required, but it can often provide interesting information that helps decide where your subsequent analysis efforts should also be aimed. In order to determine whether this file is malicious, technical indicators such as file names, hashes, strings such as network identity, domains, and data inside the header of a file can also be used. Different methods having disassemblers and network analysers may be used to observe the malware in order to collect information on how the malware functions without actually running it. Though the code is not actually executed by static perusal, sophisticated malware may inject harmful runtime activities that can go undetected. For instance, a naive static analysis could go undetected if a file constructs a string and then downloads a malicious file based on the dynamic string. For more complete knowledge of the actions of the file, organizations have moved to dynamic analysis.

6.2 Dynamic Analysis

This is the way to execute the binary measurement in an autonomous environment and monitor its behaviour. This perusal process is simple to conduct and provides helpful data into the binary's operation during its run. This methodology is useful but does not disclose all of the unkind program's functionalities. In a protected manner called a sandbox, dynamic malware perusal executes mistrustful malicious code. This closed system allows security professionals to track the malware in action without the risk of letting it infect or migrate through their system's enterprise network. Dynamic perusal offers deeper visibility to hazard hunters and incident responders, enabling them to exhibit the true nature of a threat. As a secondary advantage, the time it would take to reverse engineer a file to come across the malicious code is eliminated by automatic sandboxing. The dynamic perusal provocation is that adversaries are knowledgeable, and they know there are sandboxes out there, so they have become really good at detecting them. Adversaries conceal code within them, which may remain dormant until certain standards are fulfilled, to mislead a sandbox. Only then, the subroutine operates.

6.2.1 Cuckoo Sandbox

Cuckoo Sandbox [3] is an automated malware perusal framework with boundless application possibilities that are newfangled, highly scalable, and actually completely open source. Through definition, malicious files like executable file written in any source code, normal office documents, any kind of pdf files, emails and also malicious websites can be examined in virtualized environments under different kinds of operating systems like Windows, Linux, MacOS and Android. Trace API calls and normal file activities sublimate this into high-level facts and signatures that everyone can comprehend. Dump and evaluate, even when encrypted with secure socket layers, network traffic (Fig. 4).

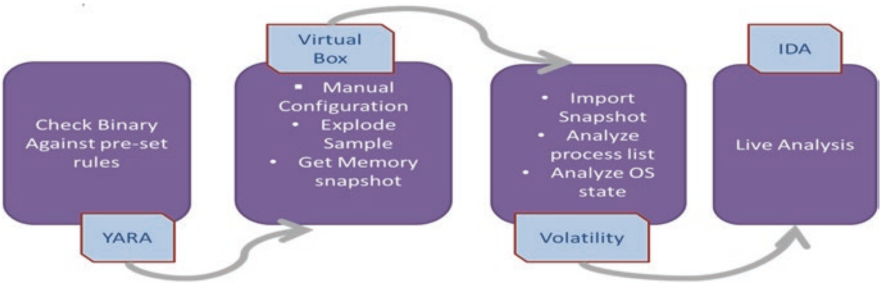


Fig. 4 Cuckoo sandbox analysis

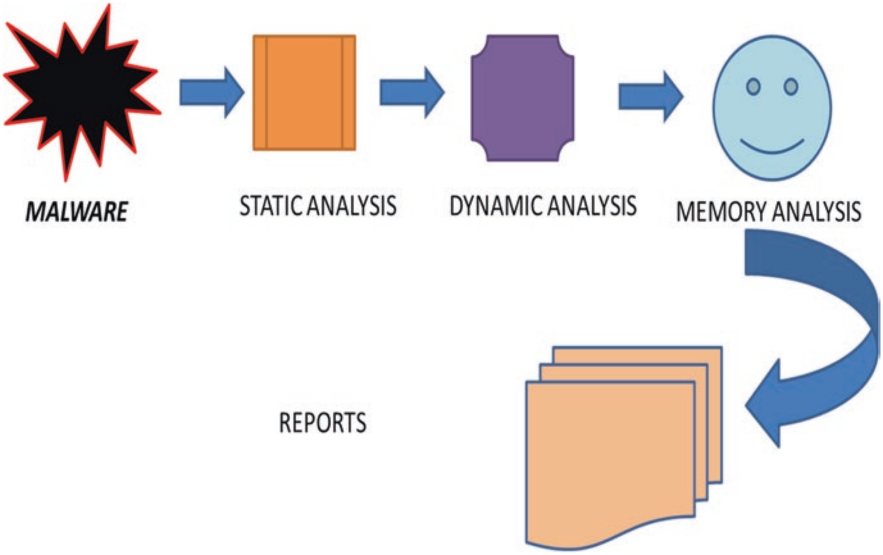


Fig. 5 Limon sandbox analysis

With the help of vernacular network routing support, it can drop all traffic or route it through InetSIM, an access point, or a virtual private network. Perform advanced storage analysis of the infected virtualized system using flickery as well as on the precision of the analyse memory through YARA. Due to the extreme available to all nature of Cuckoo and the large modular architecture, every part of the perusal setting, the processing of research results and the reporting stage can be configured. Cuckoo gives you all the specifications to quickly incorporate the sandbox according to their requirements, like one can use the sandbox in compatible format without fulfilling the licensing requirements. Because of its availability to all features, sandbox is very much used in the current environment to detect malware, and results are also satisfactory.

6.2.2 Limon Sandbox

A Limon sandbox is the most widely used product in malware detection. It basically collects the malware perform static analysis, the result of the static analysis process through dynamic analysis then performs the memory analysis feature final result. Basic process of Limon sandbox is to analyse the malware in a controlled environment, check its actions, and its sub-processes to find out the nature and intent of the malware (Fig. 5).

It tries to find out the process activity of malware, its communication mechanism with files and network connection; it also completes memory perusal and stores the analysed artifacts for later perusal. Such attacks are even possible in recent wireless communication technologies such as cognitive radio sensor networks as proposed by authors in [20].

6.2.3 Hybrid Analysis

Sophisticated malicious code may be detected through the static analysis method, but this is not a secure way; sometimes, sophisticated malware can escape detection from sandbox technology. Merging of static and dynamic analysis methods, hybrid analysis presents the security team the best of both methods – firstly because it can detect malicious code that is trying to hide, and then can extract innumerable indicators of compromise by statically and previously unseen code. The hybrid analysis may detect new threats, even those from the most enlightened malware. The hybrid analysis does, for example, apply static analysis to information by behavioural analysis – like when a piece of malicious code executes and creates some memory changes. Dynamic perusal would detect that, and analysts would be alerted to circle back and perform basic static perusal on that memory dump (Fig. 6).

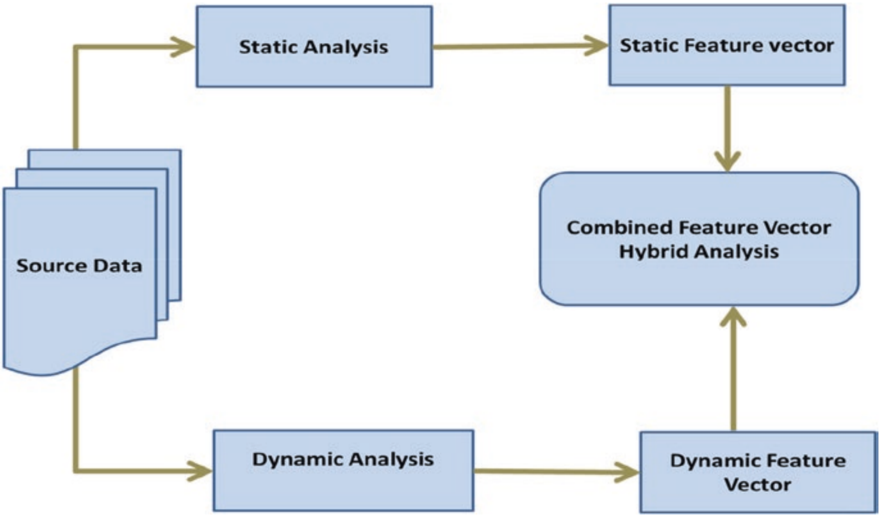


Fig. 6 Hybrid analysis process

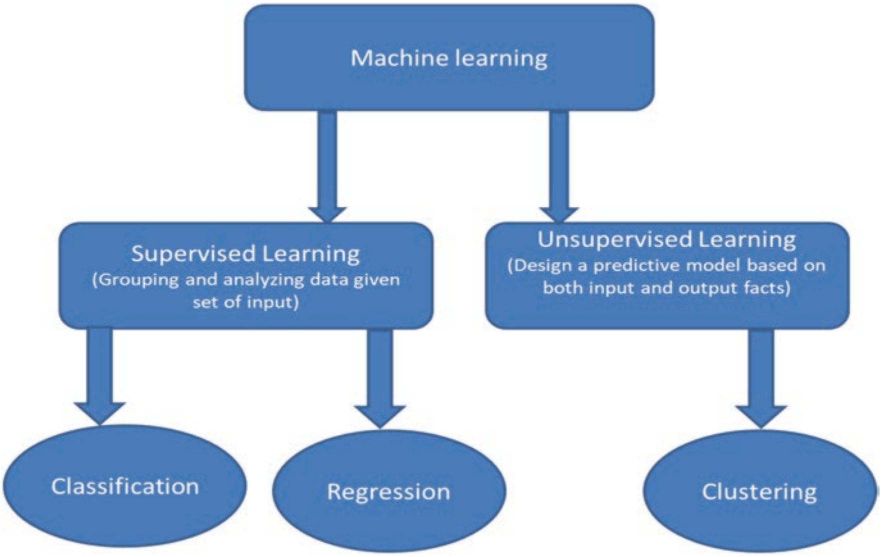

Machine learning
Supervised Learning (Grouping and analyzing data given set of input)
Unsupervised Learning (Design a predictive model based on both input and output facts)
Classification
Regression
Clustering

Fig. 7 Machine learning process bifurcation

In hybrid analysis method, the source data is processed with both static and dynamic methods. We get the static feature vector and the dynamic feature vector after processing data from both static and dynamic methods. After merging both feature vectors, we get the combined feature vector, which is also considered as hybrid analysis.

7 Machine Learning

If we think about machine learning, it takes a very technical term to listen. But if you understand about it properly, then it is very easy process which is used in almost all places nowadays. It is basically a research in computer algorithm that moves ahead by the experience. This is a type of learning in which the machine learns a lot of things without programming it explicitly. This is a type of application of AI (artificial intelligence) that gives this ability to the system so that they automatically learn from their experience and improve themselves. This may not sound true, but it is true because nowadays AI has become so advanced that it can make machines do many things which were not possible before. Since machine learning can easily handle multidimensional and multidiverse data in a dynamic environment, there are thousands of advantages of machine learning that we use in our daily work (Fig. 7).

As we have already known, machine learning is a form of artificial intelligence (AI) application that gives programs the ability to learn spontaneously and enhance themselves when needed. They use their own knowledge in order to do this, not because they are programmed directly. Computer programs development of machine learning a focus foretells at could access the data and then use it for his own learning. Learning this starts with data observations, such as direct experience or training.

It is easy to identify data trends and make informed decisions in the future. The main objective of machine learning is how to train machines automatically without any human involvement or assistance so that their actions can be modified accordingly.

7.1 Supervised Learning

A mathematical model of a group of knowledge that has both the source and, therefore, the desired targets can be developed with the study of supervised learning algorithms. In this approach, raw data can be taken as training data, and it contains many training examples. A supervisory signal is considered as a training model which has one or more inputs and, therefore, the desired output. In the mathematical model, each training example is represented by set of homogeneous elements, sometimes called a feature vector, and therefore the training data is represented by a two-dimensional array. Through loop optimization of an objective function, supervised learning algorithms learn a function which will be wont to predict the output related to new inputs. An optimal function will allow the algorithm to properly determine the output for inputs that were not a community of the training data (Fig. 8).

It is believed that an algorithm that improves the accuracy of its outputs or predictions over time has learned to perform that job.

7.1.1 The Multiple Methods of Supervised Learning

Regression

Using training data, a target outcome is generated in the method of regression. The generated outcome is a probabilistic interpretation examined after the intensity of the association between the source dataset is considered [2]. For instance,

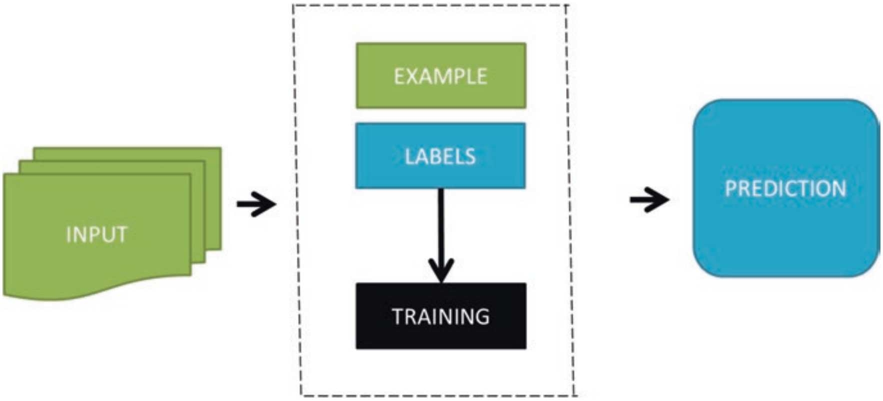


Fig. 8 Supervised learning process

regression can help predict a building's cost according to the venue, capacity, etc. The outcome has discrete values in logistic regression based on a number of independent variables.

While facing problems like non-linear and multiple decision boundaries, this approach can flounder. It is also not versatile enough for complex relationships to be recorded in datasets.

Classification

Binary Classification

It requires sorting the knowledge into grades. It is called binary classification if the supervised learning method tags input data into two separate groups. Several classifications [4] indicate the category of data into more than two different groups.

Naive Bayesian Model

An algorithm that reacts on the basis of the appearance of a certain characteristic is irrespective of the appearance of several other characteristics. The Naive Bayes algorithm's execution, considering all the sophisticated arithmetic, essentially includes keeping a list of entities with distinctive features and classes [27]. If all these statistics are collected, estimating chances and arriving at a prediction is very easy.

Decision Trees

A decision tree could be a flowchart-like design containing provisional regulation declarations, including decisions and their possible consequences. The function refers to unexpected knowledge marking. The leaf nodes refer to class labels in the tree representation, and then the internal nodes represent the features. A decision tree is sometimes used as a Boolean feature to solve issues with discrete attributes. Significant decision tree algorithms include ID3 and CART.

Random Forest Model

A prediction model is the random forest model. This involves creating a variety of decision trees and generating a hierarchy of the individual trees. Suppose you would want to guess which industry's share is going to perform well in the coming days; this algorithm will execute according to the review and previous track records of that share and current market conditions, and also will work upon the opinion of shareholders who have previously traded in these conditions; a random forest model will accomplish the objective.

Neural Networks

The purpose of this method is to combine input code, or interpret needful information by guessing and recognizing patterns. Neural networks, despite their advantages, need considerable computational resources. When there are thousands of findings, it could become difficult to suit a neural network. As analysing the logic behind their predictions is always daunting, it is often referred to as the black-box algorithm.

Support Vector Machines

A supervised learning algorithm developed in the year 1990 could be the Support Vector Machine (SVM). It draws from the hypothesis of statistical learning developed by Vap Nick. It a selective classifier because hyperplanes are removed by SVM. The efficiency is generated in the form of an appropriate hyperplane that categorizes new examples. SVMs [26] are intimately associated to and utilized in diverse sectors in the kernel framework. SVM can be used in the advancement of bioinformatics, pattern recognition, and multimedia information.

7.1.2 Pros and Cons of Supervised Learning

Many shades of supervised learning encourage peoples to accumulate and extract knowledge from previous experience. Supervised learning has demonstrated great potential in the AI domain, from optimizing recent incidents to handling actual issues. In preference to unsupervised learning, it is also a more trustworthy method, which can be computationally complex and less effective in some circumstances. Supervised learning, however, is not without shortcomings. In an IoT-based smart city architecture as implemented by authors in [21], development and progress are not possible without trust. The security of each system, sensor, and solution is not optional; it must be taken into account right from the start. For training classifiers, concrete examples are required, and decision thresholds are often overtrained in the absence of the appropriate examples. In classifying big data, one can also encounter difficulties.

7.2 Semi-Supervised Learning

The concept of semi-supervised learning comes under learning without intervention and learning with intervention. Some of the training instances are excluding training labels, yet many researchers in machine learning have found that unlabelled data can show a significant improvement in learning consistency when it is used in tandem with a minuscule percentage of labelled data. Between supervised and unsupervised learning fall semi-supervised machine learning algorithms, as they use both labelled and unlabelled data for processing, probably a decent fraction of labelled data and a big fraction of unlabelled data. The systems that use this framework are ready to improve the accuracy in learning considerably. Invariably, semi-supervised learning is chosen when skilled and specific resources are required for both the acquired labelled data to coach it/learn from it. Alternatively, it does not necessarily require substantial resources to obtain unlabelled data.

Semi-supervised learning is also very effective in the area of machine learning; its workflow is such that first the user will determine the nature of training pattern, that is, user takes decision about the training dataset which they are going to use; after the determination of training set, it needs to be related to real-world functions.

Now the next task is to find the input feature representation of the learned function; in this phase, the process will transform the input object into feature vector. Now the appropriate algorithm will be chosen for the actual execution of input data various algorithms are present like support vector machine or it may be decision trees method. Further selected algorithms should be applied on training data in order to achieve the validation set. After all these steps, finally, evaluation of the accuracy will take place and the performance of the resulting function will be measured against the test set.

These kinds of learning methods are used in different areas of application such as information extraction, spam detection and downward causation in biological system.

7.3 Unsupervised Learning

Where the information that will not be educated is neither categorized nor labelled, unsupervised machine learning algorithms are utilized. Unsupervised learning explores the way systems can infer a feature from unlabelled data to describe a secret structure. The device does not find the correct performance, but it explores the details and can draw dataset inferences to describe hidden structures from unlabelled data. Unsupervised learning algorithms take a knowledge group containing only source data and find structure within the data, such as knowledge point grouping. Therefore, the algorithms learn from raw data that has not been named, graded or categorized. Instead of responding to feedback, unsupervised learning algorithms recognize commonalities within the data and respond to each new piece of information, supporting the commonalities' existence or absence. In statistics, a core utilization of unsupervised learning is within the field of target estimation, such as discovering the function of probability density (Fig. 9).

While unsupervised learning includes several areas, data characteristics are summarized and clarified. The analysis of a cluster is the allocation of a set of observations into subsets (hierarchical clustering) in such a way that analysis within the same group are identical according to one or more predestined criteria, while analysis from different groups are different. Various clustering techniques make different assumptions about the information structure, often described by some metric of similarity and measured, for example, by internal compactness, or the similarity between members of identical groups, and the distinction between clusters. Other methods endorse the approximate density and connectivity of graphs.

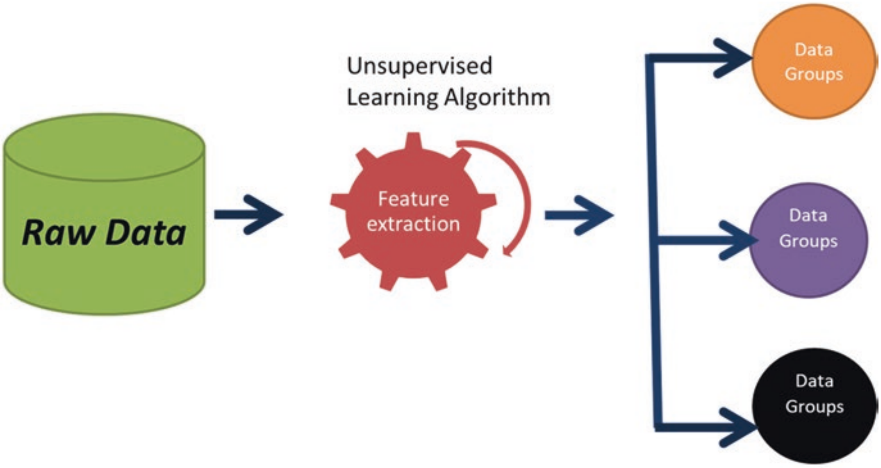


Fig. 9 Unsupervised learning process

8 Machine Learning in Malware Detection

8.1 Dataset Collection

Machine-learning classification methods need a large number of labelled executable codes for each class, but for a real-world problem such as nasty code analysis, it is very difficult to get this amount of labelled data [2]. A prolonged study process is necessary to produce these data, and some nasty executables may avoid detection within the process. Several methods are suggested to impact this problem within the full framework of machine learning. Semi-supervised learning is a type of machine-learning technique that is particularly useful for each class, if there is a limited amount of labelled data. These techniques generate a supervised classifier based on labelled data and guess the label for all unlabelled instances. Instances whose classes with a specific confidence level are expected are added to the named dataset. If such conditions are met, the procedure is repeated. The accuracy of completely unsupervised methods is enhanced by these approaches.

8.2 Features Extraction

Feature extraction [16] is a catchall concept for techniques to create correlations to fix these issues while still representing the information with adequate accuracy. Numerous machine learning experts assume that the secret to successful template matching is perfect strategy feature extraction.

8.3 Features Selection

The primary and most vital step in the current model design should be feature selection and data cleaning. You will find feature selection strategies in this chapter that you can simply use in machine learning. Feature selection [16] is the method in which you pick those features manually or automatically that most benefit your prediction variable or output during which you are curious about having irrelevant features in your data will reduce the prediction's performance and make model learn extracted attributes that are assisted.

8.4 Training of Classifier

A classification model of labelled data and appropriate assigned classification labels is acquired in one typical implementation, and an automatic classifier against training set is trained. At least one of the training samples is selected and demands confirmation/relabelling of it. A reaction identification label is obtained in response and is used to retrain the automatic classifier [23, 24].

9 Conclusion

As we found in this research, machine learning is an essential tool to combat malware. Good knowledge of machine learning and today's problems should be a distant complement.

We can avoid new and dangerous malware with the help of machine learning. Our main objective was to find a system for machine learning that typically detects the maximum number of malware samples it can, with the tough restriction of having a void false positive rate. There is a need to incorporate several deterministic exemption mechanisms in order for this system to be a part of a competitive private enterprises product. Malware detection through machine learning, in our opinion, will not replace the methods of quality detection used by anti-virus providers, but will be an additional advantage to them.

References

1. Santos, I., Nieves, J., & Bringas, P. G. (2011). Semi-supervised learning for unknown malware detection. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 415–422). Springer, Berlin, Heidelberg.
2. Anderson, H., & Roth, P. (2018). EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. ArXiv, abs/1804.04637.
3. https://cuckoosandbox.org/
4. Narudin, F. A., Feizollah, A., & Anuar, N. B. (2016). *A gani – Soft computing*. Springer.
5. Santos, I., Devesa, J., Brezo, F., Nieves, J., & Bringas, P. G. (2013). Opem: A static-dynamic approach for machine-learning-based malware detection. In International joint confer-

ence CISIS’12-ICEUTE´ 12-SOCO´ 12 special sessions (pp. 271–280). Springer, Berlin, Heidelberg.

6. https://www.quickheal.co.in/threat-reports
7. Yang, C., Xu, J., Liang, S. et al. DeepMal: maliciousness-Preserving adversarial instruction learning against static malware detection. Cybersecur 4, 16 (2021).
8. Talukder, Sajedul. (2020). Tools and Techniques for Malware Detection and Analysis.
9. Babaagba, K. O., & Adesanya, S. O. (2019). A Study on the Effect of Feature Selection on Malware Analysis using Machine Learning. In ICEIT 2019: Proceedings of the 2019 8th International Conference on Educational and Information Technology (51–55). https://doi.org/10.1145/3318396.3318448
10. Shalaginov, A., Banin, S., Dehghantanha, A., & Franke, K. (2018). Machine learning aided static malware analysis: A survey and tutorial. In Cyber threat intelligence (pp. 7–45). Springer, Cham.
11. https://symantec-enterprise-blogs.security.com/blogs/feature-stories/symantec-security-summary-june-2020
12. Hausken, K., & Welburn, J. W. (2020). *Information systems Frontiers*. Springer.
13. Kumar, M., Punia, S., Thompson, S., Gopal, D., & Patan, R. (2020). Performance analysis of machine learning algorithms for big data classification. *International Journal of E-Health and Medical Communications (IJEHMC), 12*(4), 60–75.
14. Sharma, A., & Sahay, S. K. (2014). Evolution and detection of polymorphic and metamorphic malware: A survey. *International Journal of Computer Applications, 90*(2), 7–11.
15. Govindaraju, A. (2010). *Exhaustive statistical analysis for detection of metamorphic malware*. Master’s project report, Department of Computer Science, San Jose State University.
16. Ahmadi, M., Ulyanov, D., Semenov, S., Trofimov, M., & Giacinto, G. (2016). Novel feature extraction, selection and fusion for effective malware family classification. In *ACM conference data application security privacy* (pp. 183–194). ACM.
17. Sharma, A., & Sahay, S. K. (2016). An effective approach for classification of advanced malware with high accuracy. *International Journal of Security and Its Applications, 10*(4), 249–266.
18. Bhardwaj, A., Al-Turjman, F., Kumar, M., Stephan, T., & Mostarda, L. (2020). Capturing-the-invisible (CTI): Behavior-based attacks recognition in IoT-oriented industrial control systems. *IEEE Access*, 1. https://doi.org/10.1109/ACCESS.2020.2998983
19. Shankar, A., Pandiaraja, P., Sumathi, K., Stephan, T., & Sharma, P. (2020). Privacy preserving E-voting cloud system based on ID based encryption. *Peer-to-Peer Networking and Applications*. https://doi.org/10.1007/s12083-020-00977-4
20. Stephan, T., Al-Turjman, F., Suresh Joseph, K., & Balusamy, B. (2020). Energy and spectrum aware unequal clustering with deep learning based primary user classification in cognitive radio sensor networks. *International Journal of Machine Learning and Cybernetics*. https://doi.org/10.1007/s13042-020-01154-y
21. Chithaluru, P., Al-Turjman, F., Kumar, M., & Stephan, T. (2020). I-AREOR: An energy-balanced clustering protocol for implementing green IoT in smart cities. *Sustainable Cities and Society*, 102254. https://doi.org/10.1016/j.scs.2020.102254
22. Yadav, S. P., Mahato, D. P., & Linh, N. T. D. (2020). *Distributed artificial intelligence: A modern approach* (1st ed.). CRC Press. https://doi.org/10.1201/9781003038467
23. Kumar, M., & Srivastava, S. (2018). Image authentication by assessing manipulations using illumination. *Multimedia Tools and Applications, 78*(9), 12451–11246.
24. Aggarwal, A., & Kumar, M. (2020). Image surface texture analysis and classification using deep learning. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-020-09520-2
25. O’Kane, P., Sezer, S., McLaughlin, K., & Im, E. G. (2013). SVM training phase reduction using dataset feature filtering for malware detection. *IEEE transactions on information forensics and security, 8*(3), 500–509.
26. Shang, F., Li, Y., Deng, X., & He, D. (2018). Android malware detection method based on naive Bayes and permission correlation algorithm. *Cluster Computing, 21*(1), 955–966.