International Journal of Medical Toxicology & Legal Medicine **Volume 27, No. 2, 2024**
e-ISSN: 0974-4614
p-ISSN: 0972-0448

An Attribute Similarity Based Feature Vector Training for Malware Analysis and Detection in Cloud Environments

Sanaboyina. Madhusudhana Rao[1*], Arpit Jain[2]

[1]Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India,
Email: smadhusudhan780@gmail.com
[2]Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.
*Corresponding Author

Received: 15.08.2024 Revised: 20.09.2024 Accepted: 15.10.2024

ABSTRACT
Cloud computing is not only efficient, scalable, and flexible, but it also offers a high level of reliability on elastic resources. The IT industry makes extensive use of the platform to underpin IT infrastructure and services. One of the biggest security concerns, however, is malware attacks; certain antivirus scanners aren't able to pick up on metamorphic or encrypted malware because of the environment's complexity and scale, thus these threats can get through them. Another wave with malware attacks, this time encompassing intelligent embedded devices, has arrived with the rise of the Internet of Things (IoT). Running a full malware scanner on these devices is difficult due to the low energy resources. There is a pressing need for innovative methods of scanning mobile devices for malicious software. One service that can be offered as a cloud-based option is malware detection. Dynamic behavior-based strategies have been analyzed to facilitate the analysis of possibly dangerous software. Because the executions settings in which these tactics are carried out are artificial and do not faithfully mirror the contexts of end-users, the intended recipients of the malicious activity, they sometimes produce partial results. A novel approach using Attribute Similarity based Feature Vector Training (ASbFVT) model is proposed to facilitate more accurate behavior based analysis of potentially malicious software using the extracted and selected features in the cloud environment. This platform lets users outsource tasks like program execution and analysis to remote environments like cloud servers or security laboratories, while still maintaining control over how their nodes behaves. The evaluation showed that the proposed framework enables security labs to increase the thoroughness of the analysis through carrying out a fine-grained assessment of the behavior of the program without incurring any computational cost to end-users. The proposed model when contrasted with the traditional models performs better in feature vector generation.

Keywords: Cloud Computing, Malware, Anti-Virus, Internet of Things, Security, Feature Extraction, Attribute Similarity, Feature Vector.

1. INTRODUCTION

When it comes to archiving data and offering web-based services, cloud computing is indispensable. It has many benefits over the conventional methods of storing and distributing data, including convenience, on-demand storage, scalability, and lower costs. As the IoT and Cyber-Physical Systems (CPS) become increasingly important in people's daily lives, the use of these technologies to safeguard them against cyberattacks is a top priority. The use of cloud settings to identify malware can be an effective strategy, given the exponential growth of harmful software and the lack of a widely accepted technique to its detection. Malware has evolved, and the latest strains use sophisticated obfuscation and packing methods to evade security measures.In order to penetrate and potentially destroy a computer system without the owner's knowledge, malicious software was developed. Malware is an umbrella term for any malicious software. File infectors and standalone malware make up the two main categories of malicious software. Malware can also be categorized by the harm they cause, such as worms, backdoors, trojans, rootkits, spyware, adware, and so on. Standard, signature-based methods of detecting malware are becoming increasingly ineffective, as all modern malware applications typically employ multiple distinct layers to evade detection, or use side processes to automatically update to a newer version after relatively short intervals.

Many would argue that cloud computing in its various forms has become the norm in today's society, given the pervasiveness of the Internet. Because of the importance of cloud computing to this study, a precise definition of the term is called for. The Internet is the unifying factor in many different definitions. The term cloud computing refers to the practice of employing multiple computers in a single location to do a variety of tasks

that would normally need multiple machines. Sharing computing resources and having local servers run apps is also part of cloud computing. Users of cloud computing systems need not concern themselves with data storage and transfer speeds. They can begin using the services whenever they like. The malware injection process in cloud is shown in Figure 1.

Fig 1: Malware Injection in Cloud

Malicious software is becoming a lucrative industry as it is used for spamming, committing web fraud, stealing personal information, and many other illegal activities. This financial incentive for malware development has led to a trend of increasing specialization and complexity in malicious software, with attacks increasingly targeted at narrow subsets of users and systems and constantly evolving code that incorporates new features and subtle tweaks to foil analysis and avoid detection. Security companies and academics are shifting towards adaptive behavior-based solutions to address these new threats and overcome the limits of classic malware analysis and detection methodologies. Automatically understanding the behaviors that characterize each new piece of malware and developing the relevant countermeasures is becoming the key way for security laboratories. Users can also use this technology on their own hosts to keep monitoring for dangerous processes and prevent them from running. There are two fundamental drawbacks of dynamic behavior-based analysis: incompletion and non-trivial run-time overhead.

Security laboratories conduct automated analyses of new harmful programs in specialized environmentsthat permit extremely granular monitoring of the programsbehaviours. Because the dangerous behaviours of specialized malware only appear in very specific contexts, automatic behavioural analysis of such malware is becoming increasingly challenging. The results of a behavioural study are highly likely to be inaccurate if it is conducted in unnatural settings, such as those seen in security labs. On the other hand, the malicious behaviour would have a greater probability of being triggered and caught as it manifests if the malicious program were analyzed directly on an enduser's machine, which is the intended target of the assault. Unfortunately, a fine grained examination of the behaviours of the programs is not possible due to the severe lightweight limitation necessary for end-users' systems. As a result, harmful activities may go undetected on consumer devices.

Existing approaches deal with dynamic analysis's shortcomings by exhaustively investigating every possible program route dependent on the environment. In this research, we offer a new cloud-based framework to facilitate malware analysis by combining the computational capacity available in security laboratories with the diversity of end-users' surroundings. The following two assumptions form the basis of the framework. To begin, the security lab's computational resources are unbounded, allowing it to take full advantage of hardware features and cutting-edge scientific developments to significantly enhance its computational capabilities. Second, analyzing possibly malicious software in end-users' surroundings is preferable than doing so in the synthetic environments normally accessible in security labs since they are more realistic and varied. Using the suggested framework, a user can hand off the task of running and analyzing a suspicious program to a security lab, while nevertheless imposing the same constraints on the program as if it were being run locally. The different malware detection types in cloud environment are shown in Figure 2.

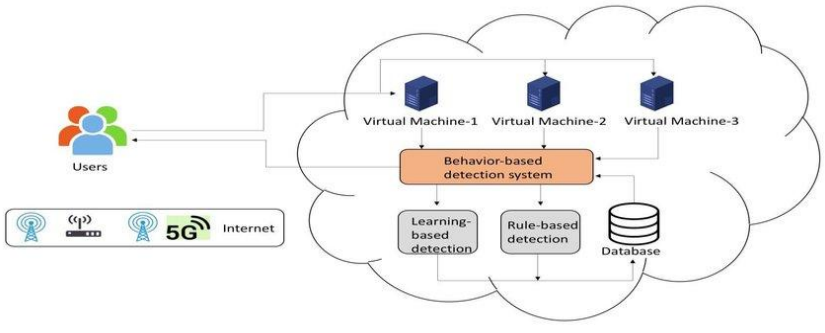

Fig 2: Malware Detection Types

By leveraging the computing capabilities of the security lab, end users can increase their level of protection and the lab can monitor the execution of a potentially malicious program in an environment that is realistic for the end user. This framework allows the security lab to increase the thoroughness of the analysis by watching how a program operates in a number of realistic end-user environments, which is important because each user has a slightly different setup. The mechanism for transmitting the system calls that the analyzed program makes to a distant end-user's environment for execution and returning the results of the calculation enables this kind of execution in the cloud. Given that a program's execution flow is determined solely by the results of the system calls it makes, an analyzed program operating in a security lab appears to behave as though it were being performed in the user's actual environment.A novel approach using Attribute Similarity based Feature Vector Training model is proposed to facilitate more accurate behavior based analysis of potentially malicious software using the extracted and selected features in the cloud environment.

2. LITERATURE SURVEY

When transferring private information to the cloud, encryption is a must. When it comes to protecting the confidentiality and integrity of data stored in the cloud, Attribute-Based Encryption (ABE)-based access control is one of the most efficient methods available. Since the ABE Cipher-Text Policy does not allow for the addition or deletion of computing nodes at run time, it may have scalability and performance concerns. Single-point-of-failure (SPoF) is another issue with current methods. Therefore, in our proposed work, Qaisar et al. [1] introduced a scalable multi-agent system structure based on CP-ABE to guarantee data sharing on public cloud storage with high levels of availability. Without compromising the privacy and security of the system, the author proposed using a cloud host as an intermediary between the user and the authorised agents. The author also presented a new strategy to cloud security, one that makes use of the efficient power of the state-of-the-art Gemini technique to counteract malware. In order to find similarities across graph embeddings using only binary codes, Gemini is a useful strategy. This suggested research provides an approach for malware detection in the cloud, while also addressing issues with scalability and efficiency. It addresses three key areas: scalability, multi-agent efficiency, and malware detection.

The use of cloud computing is one of the more promising technologies now available for storing data and delivering services via the Internet. There are significant benefits to be gained over more conventional protection methods by making use of this fast evolving technology to shield computer-based systems from cyber-related threats. Cyber-physical systems (CPS), critical systems, conventional computers, portable computers, mobile devices, and the IoT are all examples of the types of computer-based systems that need to be safeguarded. Malicious software (malware) refers to any program that intentionally compromises a computer system in order to compromise its security, privacy, or availability. Aslan et al. [2] proposed a cloud-based intelligent behavior-based detection system to identify the ever-increasing malware attack surface. The suggested approach begins by collecting malware samples from a variety of virtual machines, from which unique characteristics can be extracted easily. To distinguish malicious from safe files, the author next feed selected attributes to our learning-based and rule-based detection agents. The effectiveness of the suggested approach was assessed by analysing 10,000 samples of code. The suggested method has a high rate of detection and accuracy for both known and undiscovered malware.

The rapid development of the industrial Internet has led to the emergence of cloud service as a cutting-edge industrial norm with game-changing possibilities for the business world. Many companies have come to see the value of cloud-based service models in recent years. However, there is a huge worry about security vulnerabilities, such as malware attacks on virtual domains that go undetected. To combat malware in cloud infrastructure, we offer an introspection-based security method (named VMShield) for protecting virtual domains within a cloud-based service platform. VMShield prevents malware from evading the security tool by performing virtual memory introspection from the hypervisor (trusted-domain) to capture the behaviour of processes in real time. The proposed method by Mishra et al. [3] was superior to both static and dynamic state-of-the-art procedures because of its reliance on introspection to uncover stealthy assaults. The VMShield uses the Bag of n-gram method to extract features from system calls and the meta-heuristic methodology binary particle swarm optimisation to choose relevant features. The observed programmes are divided into good and bad processes using a Random Forest (RF) classifier, which may detect variations of malware more effectively than the standard signature-matching method.

To support their many different software programs, many businesses are turning to cloud computing. Businesses can reduce costs associated with hardware management, scalability, and maintenance thanks to these services. Amazon Web Services, Microsoft Azure, and Google Cloud Platform are just a few of the leading CSPs that provide Infrastructure as a Service (IaaS) to businesses like these. Because of this, CSPs must prioritise the safety of their cloud services because it has become a prime target for cybercriminals as cloud usage rises. Malware is widely acknowledged as one of the most severe and pervasive dangers to IaaS in the cloud. In this paper, Kimmel et al. [4] investigated how well cloud VMs can identify malware using deep learning approaches based on Recurrent Neural Networks (RNNs). The author zeroed in on the Long Short Term Memory RNNs

(LSTMs) and the Bidirectional RNNs (BIDIs) as the primary RNN architectures of interest. Malware behaviour is learned by these models over time by observing system parameters like CPU, memory, and disc utilization at runtime during malicious attacks. This method is tested on a dataset consisting of 40,680 harmful and benign samples. Essential for simulating realistic cloud provider circumstances and for accurately capturing the behaviour of stealthy and sophisticated malware, the process level features were collected using actual malware running in an open, online cloud environment with no constraints.

Due to the severity of the threats posed by Android malware, the need to identify it has become critical. Privacy issues and communication bottlenecks are commonplace in in-cloud Android malware detection. In actual use, the need for offline updates highlights the need of on-device training. However, on-device training is challenging to accomplish, especially for those high-complexity malware detectors, due to the restricted resources of mobile devices. Using the recently described wide learning method, Yua et al. [5] developed a lightweight on-device Android malware detector to address this issue. In order to train its models, our detector mostly use one-time computing. As a result, it may be trained in its entirety or in stages, right on the go with a mobile device. The detection accuracy of our detector is comparable to that of the deep learning-based models multilayer perceptron (MLP) and convolutional neural network (CNN), while it surpasses the shallow learning-based models support vector machine (SVM) and AdaBoost. In addition, compared to current detectors, ours can better withstand hostile cases, and this is only going to become better with on-device model retraining.

In light of recent advancements in computer systems, people are increasingly spending more time in simulated worlds. This process has been sped up by the Covid-19. The focus of cybercriminals has switched from the actual world to the online world. This is due to the fact that it is less difficult to engage in criminal activity online compared to the real world. To launch cyberattacks, cybercriminals commonly employ malicious software, sometimes known as malware. Advanced obfuscation and packing techniques are being included into new malware versions. Malware identification and categorization are greatly complicated by these methods of concealment. To effectively tackle emerging malware strains, novel ways that are very different from conventional methods must be applied. Traditional machine learning (ML) techniques, a subset of AI, are unable to detect all of today's sophisticated malware. The challenge of detecting new forms of malware requires a novel methodology, and the deep learning (DL) method offers hope in this regard. In this research, Aslan et al. [6] presented a new deep-learning-based architecture for malware classification that makes use of a hybrid model. The primary result of this research is anovel hybrid architecture for optimally combining two diverse pre-trained network models. The four primary steps of this architecture include gathering data, planning the structure of the deep neural network, training the network, and then assessing its performance.

Containerization's popularity in the cloud has been on the rise recently. Kubernetes has made it easier to control software running in isolated containers. Kubernetes makes it possible to automate processes related to application administration, such as self-healing, scaling, rolling back, and updating. Attacks on pods to carry out harmful acts are another example of how security threats have grown. Cryptomining malware, which steals computing power to mine bitcoin, is among the most dangerous new strains of malware. A cryptomining process, initiated by a concealed malware executable, can operate in the background during application deployment and operation in the pod; thus, a way to detect malicious cryptomining software running inside Kubernetes pods is required. One approach that could work is to utilise machine learning (ML) to determine if a pod is hosting a cryptomining operation and then label it accordingly. In addition to detection, the system administrator will require an explanation of the ML's classification outcome and its justification. A pod's removal or restart with a new image are both disruptive administrative actions, and the explanation will justify and support these actions. Karn et al. [7] detailed the planning and execution of a machine learning (ML)-based system for detecting anomalous pods in a Kubernetes cluster by monitoring the scalls made by the Linux kernel. In an anomalous pod, many cryptominer image containers are employed, and multiple ML models are developed to identify these pods among the many normal cloud workloads.

The advent of 5G is laying the groundwork for the IIoT by facilitating the low-latency integration of AI and cloud computing within the context of a smart and intelligent IIoT ecosystem, which in turn improves the overall industrial process. However, it also provides new powerful attack vectors, which poses serious security and data privacy threats, and raises the functional difficulties of the underlying control system. Security and privacy are becoming increasingly important as malware begins to target vulnerable yet highly connected IoT devices. In order to categorise malware attacks on the IIoT, Ahmed et al. [8] proposed a 5G-enabled system consisting of a deep learning-based architecture. This approach makes use of convolutional neural networks trained to distinguish between distinct types of malware attacks based on a visual representation of the virus.

3. PROPOSED MODEL

Malicious software, or malware, is computer code that was created with malicious goals. Malware's primary function is to invade users' privacy and steal their personal information, which has spurred a constant evolution of malevolent tactics. However, security researchers represent a particularly serious hazard. Traditional methods, such as signature-based and static malware analysis, are still frequently used, despite requiring human

analysts to analyze the harmful code by hand. Unfortunately, due to the exponential rise in malware production, antivirus companies now have to deal with thousands of new malware files every day. Due to obfuscation, polymorphism, etc., this method is unable to analyze a vast volume of files. Therefore, a trustworthy and automated analysis is crucial for dealing with this danger. The latest generation cyber threats/attacks are becoming more targeted, persistent, and unknown as a result of the increased availability and sophistication of tools. When compared to classic malware, which was widespread, well-documented, and used just once, modern malware is targeted, unknown, stealthy, personalised, and contains zero days. They get inside, multiply, and weaken the host's defences.

Antivirus (AV) software was developed to help stop, identify, and delete viruses and other forms of malware. Automatically categorizing and analyzing malware samples was the primary goal of this project. Unfortunately, polymorphism and metamorphic strategies make these methods fundamentally inaccurate. So, neither the IDS nor the antivirus were fully successful in their study of malware. As a result, malware samples are often automatically classified and categorized using dynamic malware analysis techniques. Execution of binaries in a sandboxed setting to glean the information needed to spot malicious behaviour is at the heart of dynamic malware analysis. In recent years, cloud infrastructure has been increasingly popular due to the advantages it offers in terms of scalability, adaptability, and cost-effectiveness. The cloud is becoming increasingly prevalent in the modern era, yet it is still vulnerable to serious attacks from cybercriminals. In a virtualized setting, security and dependability are paramount. Clouds, however, have hampered the conventional detection method due to the nature and operational research structures of clouds. The proposed model framework is shown in Figure 3.

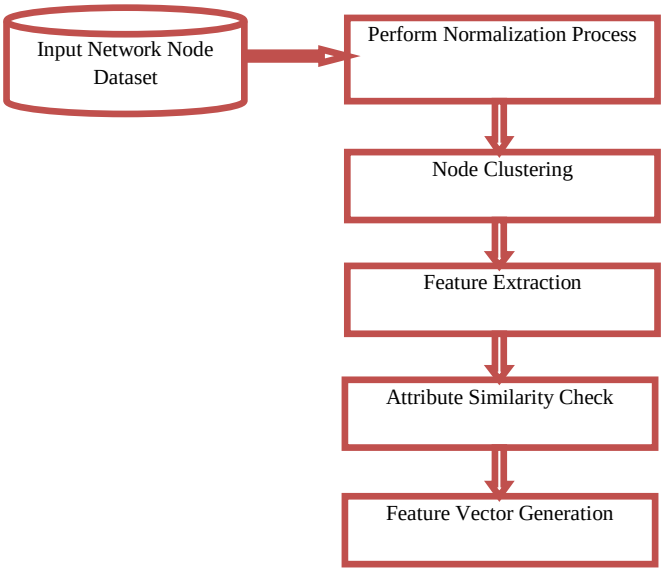

Fig 3: Proposed Model Framework

Training and testing are the two main components of Malware Detection methods. The entire procedure has two distinct phases. The first is the training phase, and the second is the evaluation phase. In the Training phase, features are taken from both harmful and benign applications/files and then used to teach the classifier to correctly identify whether an input file is dangerous or benign. The learned classifier is then utilized to determine whether or not the entered file is harmful during the testing phase. The method of Feature Extraction is crucial to Malware Detection. In this process, we only select those traits that may reliably provide useful discriminatory data. It has significant effects on the algorithm's precision and performance. The goal of Feature Extraction is to identify a transformation, typically linear, that can better separate classes by doing away with extraneous information and redundancies. Inefficient file detection may occur due to the enormous collection of features provided. So, in pattern recognition and categorization problems, where just the well-chosen features offer discrimination information, Feature Extraction is performed to provide efficient and accurate file detection that can reveal even the minutest of changes in the machine's condition. In this sense, feature extraction can be thought of as a type of feature selection that also involves a transformation. In most cases, reducing the feature space and cutting down on measurement costs are the primary goals. After extracting all the features from the dataset, feature selection strategy is applied on the extracted features to consider the most relevant features for training the model to detect malware in the cloud environment. A novel approach using Attribute Similarity based Feature Vector Training (ASbFVT) model is proposed to facilitate more accurate behavior based analysis of potentially malicious software using the extracted and selected features in the cloud environment.

Algorithm ASbFVT

{

Step-1: Initially load the dataset and analyze each record from the dataset for malware detection. The records in the dataset are analyzed as

$$NodeReg\quad[N]=\sum_{i=1}^{N} nodebaseaddr\quad(i)+getloc\quad(i)+\frac{alloc\quad(enerLevel\quad(i))}{\lambda}+Th$$

Here nodebaseaddr() considers the node address, getloc() is used to identify the location of nodes and allocated energy is considered for each node and λ is the total energy of the network. Th is the threshold value.

Step-2: Each node records are processed using the standard mean application to detect the relevant properties in the network. The node normalization helps in considering of values that are in range to the normal and malicious actions. The normalization process is performed as

$$Norm\quad[N]=\prod_{i=1}^{N} getattr\quad(NodeReg\quad(i))+mean\quad(i,i+1)+std\quad(i,i+1)+\min(getattr\quad(i,i+1)+\frac{maxattr\quad(i,i+1)}{\sqrt{len\quad(NodeReg\quad)}}$$

Here getattr() is used to retrieve each value from the node registration data, mean() is used to calculate the mean among the two adjacent attributes i and i+1. Std() is used for calculating the standard deviation.

Step-3: After normalization, each record is analyzed and the clustering process is performed. Clustering is the process of maintaining similar kind of values as a group. The node clustering is performed based on node attributes that is performed as

$$NodeClus\quad[N]=\sum_{i=1}^{N} getattr\quad(i)+\frac{simm\quad(Norm\quad(i,i+1))}{\gamma}+\max(attr\quad(Norm\quad(i,i+1))\begin{cases}NodeClus\quad\leftarrow attr\quad(i)\ if simm\quad(Norm\quad(i,i+1))>\omega\\No\ deClus\quad\leftarrow attr\quad(i+1)\ if simm\quad(Norm\quad(i,i+1))<\omega\end{cases}$$

Here simm() is used to calculate the similarity levels among two attributes. γ is the total attributes count in Norm set. ω is the threshold value for forming clusters.

Step-4: Perform extraction of features from the dataset and use these features for training the model. The feature extraction considers all the features from the dataset and then the high priority features in detecting intrusions are considered for detecting intrusions. The feature extraction process is performed as

$$Featextr\quad[N]=\sum_{i=1}^{N}\frac{getattr\quad(i)}{\gamma}+\max(attr\quad(i,i+1))-\min(attr\quad(i,i+1))+\lim_{i\to N}\left(\gamma+\frac{\omega}{i}\right)^{2}$$

Step-5: Classify the features based on the attack set and generate a feature subset based on attack type. Each feature subset is used for detecting an attack type. The feature subset generation is performed based on the attribute similarity range. The attribute similarity measurement is performed as

$$AttrSimm\quad[N]=\prod_{i=1}^{N}\frac{\max(getattr\quad(i,i+1))}{\gamma}+\frac{\min(getattr\quad(i,i+1))}{\gamma}+\text{simm}(\frac{\max(i),\min(i))}{\omega})\begin{cases}\text{simm}\leftarrow 1 \text{ if simm}(\max(i),\min(i))>\omega\\\text{simm}\leftarrow 0 \text{ if simm}(\max(i),\min(i))<\omega\end{cases}$$

Step-6: The final feature vector set is generated in which the vector is used to detect the malware in the cloud environment. The feature vector generation is performed as

$$FeatVec\quad[N]\sum_{i=1}^{N}\frac{getmax\quad(AttrSimm\quad(i,i+1))}{\gamma}+\max(corr\quad(AttrSimm\quad(i,i+1))-\min(corr\quad(AttrSimm\quad(i,i+1)+\frac{len\quad(AttrSimm\quad)}{\gamma}$$

Here corr() is used to find the relational factor among two adjacent features. The features that are related will be selected in the final feature vector.

}

4. RESULTS

In recent years, cloud computing has emerged as the dominant computer model. More and more services are being moved to the cloud because of the scalability and adaptability it provides. When it comes to cloud infrastructure, however, the cloud provider has complete authority but lacks the contextual understanding of the hosted virtual machines.There are two types of data used in the experiment: training data and test data. The algorithms used to learn the characteristics of malware samples are first provided with training samples. A dynamic analysis of these malware samples provides descriptive facts about the actions taken during the analysis, vividly portraying the many features including network, file system, registry, and so on. A novel approach using Attribute Similarity based Feature Vector Training (ASbFVT) model is proposed to facilitate more accurate behavior based analysis of potentially malicious software using the extracted and selected features in the cloud environment. The proposed model is compared with the traditional Scalable and Efficient Multi-Agent Architecture for Malware Protection (SEMAMP) model and Intelligent Behavior-Based Malware Detection System on Cloud Computing Environment (IBMDS). The proposed model when compared with the traditional models, the proposed model performance is high in generation of relevant feature vector.

One of the most typical methods of data preparation is normalization, which allows users to convert the numerical values of each column in the dataset to a standard scale. The purpose of normalization is to adjust feature values such that they are all roughly the same. The model's performance and stability throughout training are both enhanced as a result. To ensure that all numerical columns in the dataset are measured on the same scale, normalization is a data preparation step in machine learning that employs a scaling algorithm. Only when aspects of machine learning models fall into varying categories does this step become necessary. The feature normalization process reduces the range of the feature values to a minimum scale range. The Feature Normalization Accuracy Levels of the proposed and existing models are shown in Table 1 and Figure 4.

Table 1: Feature Normalization Accuracy Levels

Records Considered	Models Considered		
	ASbFVT Model	SEMAMP Model	IBMDS Model
15000	97.4	91.4	93.3
30000	97.6	91.7	93.6
45000	97.9	92	93.8
60000	98	92.3	94.1
75000	98.3	92.7	94.3
90000	98.5	93	94.5

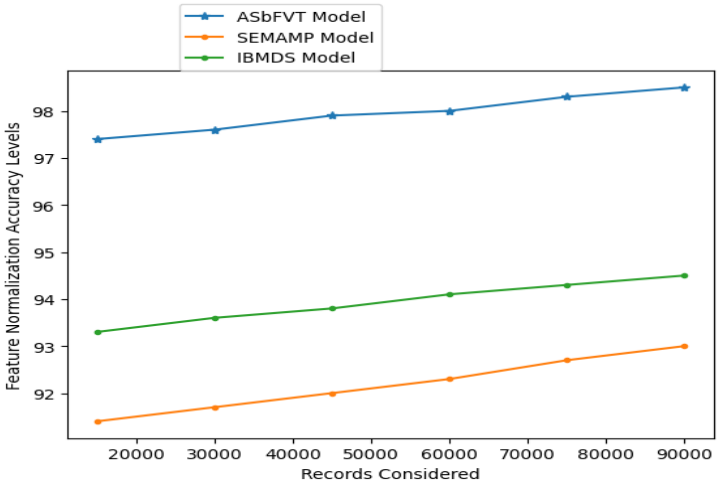

Fig 4: Feature Normalization Accuracy Levels

To extract highly interconnected but comparatively separate nodes from a cloud network, a set of algorithms and approaches known as cloud network clustering have been developed. When it comes to managing the topology of networks, clustering is a common method. In order to achieve goals like providing Quality of Service (QoS), minimizing the utilization of resources, network load balancing, etc., a clustering technique sorts nodes into groups called clusters. The Node Clustering Time Levels of the existing and proposed models are shown in Table 2 and Figure 5.

Table 2: Node Clustering Time Levels

Records Considered	Models Considered		
	ASbFVT Model	SEMAMP Model	IBMDS Model
15000	11	17	22
30000	12	18	22.5
45000	13	19	24
60000	14	20	25
75000	15	21	27
90000	16	22	28

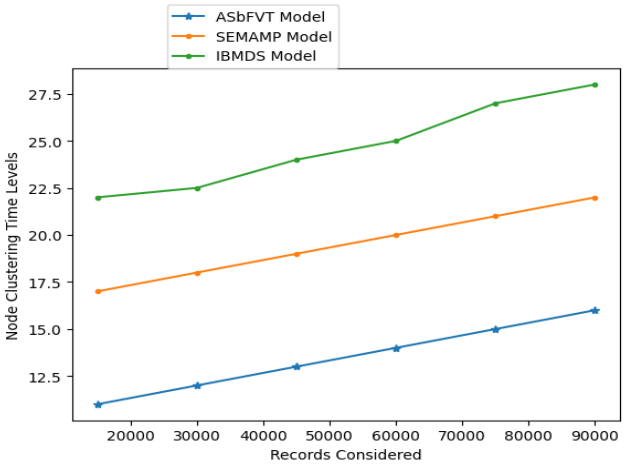

Fig 5: Node Clustering Time Levels

Data transformation into numerical characteristics that may be processed without losing any of the original data's meaning is known as feature extraction. When compared to using machine learning on unprocessed data, the results are far more favorable. By eliminating unnecessary information, feature extraction cleans up the dataset. In the end, data reduction aids model construction while reducing machine effort and accelerates machine learning's learning and generalization phases. The Table 3 and Figure 6 show the Feature Extraction Accuracy Levels of the existing and proposed models.

Table 3: Feature Extraction Accuracy Levels

Records Considered	Models Considered		
	ASbFVT Model	SEMAMP Model	IBMDS Model
15000	97.5	89	94.2
30000	97.6	90	94.5
45000	97.9	91	94.7
60000	98	92	95
75000	98.2	93	95.3
90000	98.4	94	95.5

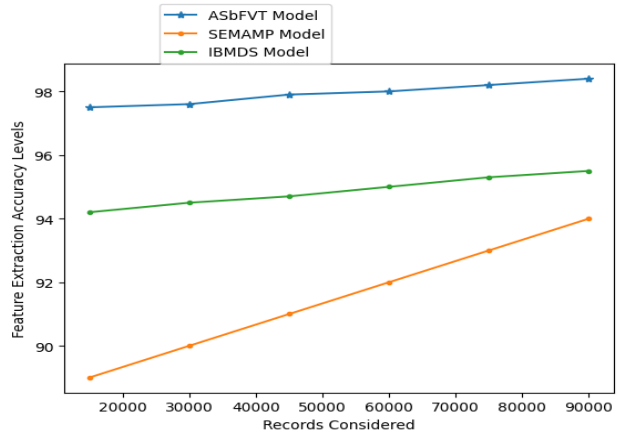

Fig 6: Feature Extraction Accuracy Levels

The feature data for two features must be combined into a single numerical number in order to determine the similarity between the two feature sets. The attribute similarity check is performed for the detection of malware in the cloud environment. The changes in the attributes results in malware. The Attribute Similarity Check Time Levels of the proposed and existing models are shown in Table 4 and Figure 7.

Table 4: Attribute Similarity Check Time Levels

Records Considered	Models Considered		
	ASbFVT Model	SEMAMP Model	IBMDS Model
15000	12.4	15.5	18.4
30000	12.7	16	18.8
45000	13	16.5	19.2
60000	13.4	17	19.8
75000	13.7	17.5	20.4
90000	14	18	21

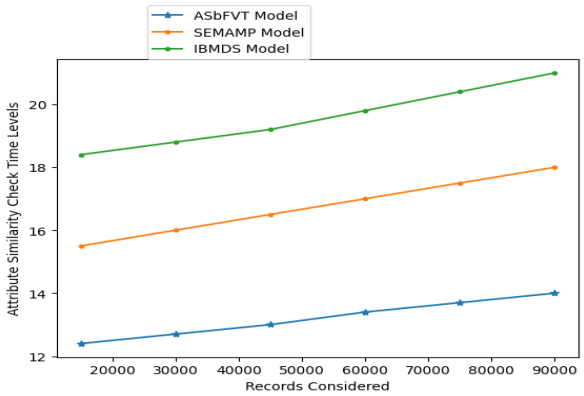

Fig 7: Attribute Similarity Check Time Levels

An ordered reduced array of numerical characteristics of an observed phenomenon is called a feature vector that contains only the most useful attributes. To a machine learning model, it stands for the features used to produce a prediction. A feature vector represents a measurable attribute of some observable phenomenon. The Feature Vector Generation Accuracy Levels of the proposed and existing models are shown in Table 5 and Figure 8.

Table 5: Feature Vector Generation Accuracy Levels

Records Considered	Models Considered		
	ASbFVT Model	SEMAMP Model	IBMDS Model
15000	97.9	93	91.5
30000	98	93.5	91.9
45000	98.1	93.8	92.2
60000	98.3	94.1	92.9
75000	98.5	94.5	93.4
90000	98.6	95	94

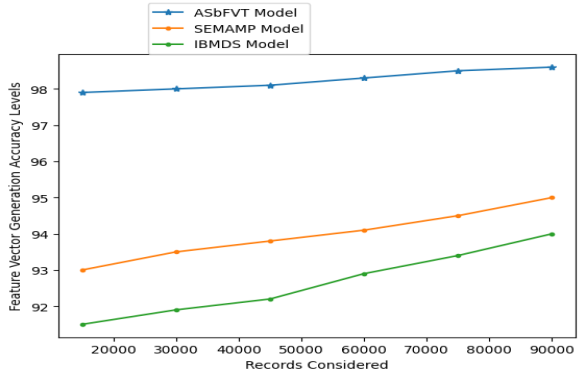

Fig 8: Feature Vector Generation Accuracy Levels

Attacks found in the considered dataset are depicted in Figure 9. Assaults and totals are shown in the dataset.

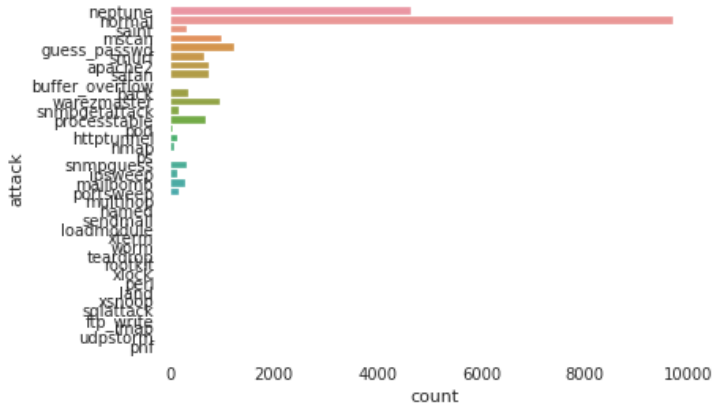

Fig 9: Attack Count

Figure 10 depicts the model's assault detection flags. The total number of flags is clearly displayed.

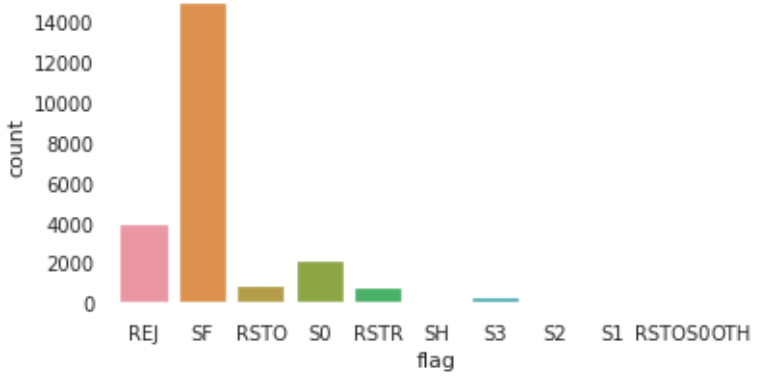

Fig 10: Flag Types Used

5. CONCLUSION

Malware is growing in prevalence as the world becomes increasingly digital. The proliferation of malware has accelerated the need for preventative measures. Existing static and dynamic malware analysis approaches have blind spots when it comes to detecting the most sophisticated threats. As a result, techniques based on memory analysis have become increasingly valuable for finding malicious software. The severity of malicious actions in the virtualized environment is growing. There are security gaps in the Internet that hackers can use to gain access to private information. The methods employed by the hackers, as well as the inability to recognize the new unknown signature of the malware, have rendered the fundamental and static approaches useless in this field. Dynamic malware analysis has shown to be the most effective of the several approaches employed thus far in the evolution of automated malware analysis. This proposed technique is helpful in the detection of malware instance in the virtualized environment by extracting malware behavior, retrieving the best feature, clustering to the relevant prototype, and assigning to the appropriate class. Finally, a system that combines malware detection systems with cloud computing environments is proposed, where all binaries and malware in use are intercepted and sent to one or more analysis engines for a full check against a signature database in order to find previously unknown exploits or malware. The primary goal of this effort is to improve malware sample clustering and classification by accurately portraying features as either harmful or safe. It can be used to counteract security flaws in virtual machines and hostile hacking attempts. Data is protected at a high rate because virtual machines are linked to the host, which is protected at the hypervisor level. A novel approach using Attribute Similarity based Feature Vector Training model is proposed to facilitate more accurate behavior based analysis of potentially malicious software using the extracted and selected features in the cloud environment. The proposed model achieves 98.6% accuracy in generation of feature vector that is used to training the model for accurate malware detection. Integration of the proposed system's modules with improved machine learning algorithms will allow for efficient clustering and categorization of malware data in the future. Feature dimensionality reduction techniques also can be applied in future to further reduce the feature set for improving the accuracy levels and reducing the training time complexity levels.

REFERENCES

1. Z. H. Qaisar, S. H. Almotiri, M. A. Al Ghamdi, A. A. Nagra and G. Ali, "A Scalable and Efficient Multi-Agent Architecture for Malware Protection in Data Sharing Over Mobile Cloud," in IEEE Access, vol. 9, pp. 76248-76259, 2021, doi: 10.1109/ACCESS.2021.3067284.
2. Ö. Aslan, M. Ozkan-Okay and D. Gupta, "Intelligent Behavior-Based Malware Detection System on Cloud Computing Environment," in IEEE Access, vol. 9, pp. 83252-83271, 2021, doi: 10.1109/ACCESS.2021.3087316.
3. P. Mishra et al., "VMShield: Memory Introspection-Based Malware Detection to Secure Cloud-Based Services Against Stealthy Attacks," in IEEE Transactions on Industrial Informatics, vol. 17, no. 10, pp. 6754-6764, Oct. 2021, doi: 10.1109/TII.2020.3048791.
4. J. C. Kimmel, A. D. Mcdole, M. Abdelsalam, M. Gupta and R. Sandhu, "Recurrent Neural Networks Based Online Behavioural Malware Detection Techniques for Cloud Infrastructure," in IEEE Access, vol. 9, pp. 68066-68080, 2021, doi: 10.1109/ACCESS.2021.3077498.
5. W. Yuan, Y. Jiang, H. Li and M. Cai, "A Lightweight On-Device Detection Method for Android Malware," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5600-5611, Sept. 2021, doi: 10.1109/TSMC.2019.2958382.
6. Ö. Aslan and A. A. Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," in IEEE Access, vol. 9, pp. 87936-87951, 2021, doi: 10.1109/ACCESS.2021.3089586.
7. R. R. Karn, P. Kudva, H. Huang, S. Suneja and I. M. Elfadel, "Cryptomining Detection in Container Clouds Using System Calls and Explainable Machine Learning," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 3, pp. 674-691, 1 March 2021, doi: 10.1109/TPDS.2020.3029088.
8. Ahmed, M. Anisetti, A. Ahmad and G. Jeon, "A Multilayer Deep Learning Approach for Malware Classification in 5G-Enabled IIoT," in IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 1495-1503, Feb. 2023, doi: 10.1109/TII.2022.3205366.
9. Yucel, C.; Koltuksuz, A. Imaging and evaluating the memory access for malware. Forens. Sci. Int. Digit. Investig. 2020, 32, 200903.
10. Banin, S.; Dyrkolbotn, G.O. Detection of Previously Unseen Malware Using Memory Access Patterns Recorded before the Entry Point. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 2242–2253.
11. Sihwail, R.; Omar, K.; Zainol Ariffin, K.A. A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis. Int. J. Adv. Sci. Eng. Inf. Technol. 2018, 8, 1662–1671.
12. Mosli, R.N.; Li, R.; Yuan, B.; Pan, Y. Automated malware detection using artifacts in forensic memory images. In Proceedings of the 2016 IEEE Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 10–11 May 2016.
13. Rathnayaka, C.; Jamdagni, A. An Efficient Approach for Advanced Malware Analysis Using Memory Forensic Technique. In Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, NSW, Australia, 1–4 August 2017; pp. 1145–1150.
14. Sihwail, R.; Omar, K.; Ariffin, K.A.Z. An Effective Memory Analysis for Malware Detection and Classification. CMC Comput. Mater. Contin. 2021, 67, 2301–2320.
15. Sihwail, R.; Omar, K.; Ariffin, K.A.Z.; Al Afghani, S. Malware Detection Approach Based on Artifacts in Memory Image and Dynamic Analysis. Appl. Sci. 2019, 9, 3680.
16. Ucci, D.; Aniello, L.; Baldoni, R. Survey of machine learning techniques for malware analysis. Comput. Secur. 2019, 81, 123–147.
17. Aghaeikheirabady, M.; Farshchi, S.M.R.; Shirazi, H. A New Approach to Malware Detection by Comparative Analysis of Data Structures in a Memory Image. In Proceedings of the 2014 International Congress on Technology, Communication and Knowledge (ICTCK), Mashhad, Iran, 26–27 November 2014.
18. Mohaisen, A.; Alrawi, O.; Mohaisen, M. AMAL: High-fidelity, behavior-based automated malware analysis and classification. Comput. Secur. 2015, 52, 251–266.
19. Ahmadi, M.; Ulyanov, D.; Semenov, S.; Trofimov, M.; Giacinto, G. Novel Feature Extraction, Selection and Fusion for Effective Malware Family Classification. In Proceedings of the Codaspy'16: Proceedings of the Sixth Acm Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 9–11 March 2016; pp. 183–194.
20. Kumara, M.A.A.; Jaidhar, C.D. Leveraging virtual machine introspection with memory forensics to detect and characterize unknown malware using machine learning techniques at hypervisor. Digit. Investig. 2017, 23, 99–123.
21. Mosli, R.; Li, R.; Yuan, B.; Pan, Y. A Behavior-Based Approach for Malware Detection. IFIP Adv. Inf. Commun. Technol. 2017, 511, 187–201.

22. Petrik, R.; Arik, B.; Smith, J.M. Towards Architecture and OS-Independent Malware Detection via Memory Forensics. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (Ccs'18), Toronto, ON, Canada, 15–19 October 2018; pp. 2267–2269.
23. Nissim, N.; Lahav, O.; Cohen, A.; Elovici, Y.; Rokach, L. Volatile memory analysis using the MinHash method for efficient and secured detection of malware in private cloud. Comput. Secur. 2019, 87, 101590.
24. Lashkari, A.H.; Li, B.; Carrier, T.L.; Kaur, G. VolMemLyzer: Volatile Memory Analyzer for Malware Classification Using Feature Engineering. In Proceedings of the 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), Hamilton, ON, Canada, 18–19 May 2021; pp. 1–8.
25. Severi, G.; Leek, T.; Dolan-Gavitt, B. MALREC: Compact Full-Trace Malware Recording for Retrospective Deep Analysis. In Detection of Intrusions and Malware, and Vulnerability Assessment; Springer: Cham, Switzerland, 2018; Volume 10885, pp. 3–23.
26. Kang, J.; Jang, S.; Li, S.; Jeong, Y.S.; Sung, Y. Long short-term memory-based Malware classification method for information security. Comput. Electr. Eng. 2019, 77, 366–375.
27. Safa, H.; Nassar, M.; Al Orabi, W.A. Benchmarking Convolutional and Recurrent Neural Networks for Malware Classification. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 561–566.
28. Lu, X.F.; Jiang, F.S.; Zhou, X.; Yi, S.W.; Sha, J.; Lio, P. ASSCA: API sequence and statistics features combined architecture for malware detection. Comput. Netw. 2019, 157, 99–111.
29. Sung, Y.; Jang, S.; Jeong, Y.S.; Park, J.H. Malware classification algorithm using advanced Word2vec-based Bi-LSTM for ground control stations. Comput. Commun. 2020, 153, 342–348.
30. Panker, T.; Nissim, N. Leveraging malicious behavior traces from volatile memory using machine learning methods for trusted unknown malware detection in Linux cloud environments. Knowl. Based Syst. 2021, 226, 107095.
31. Diaz, J.A.; Bandala, A. Portable Executable Malware Classifier Using Long Short Term Memory and Sophos-ReversingLabs 20 Million Dataset. In Proceedings of the TENCON 2021—2021 IEEE Region 10 Conference (TENCON), Auckland, New Zealand, 7–10 December 2021; pp. 881–884.
32. Wang, Q.H.; Qian, Q. Malicious code classification based on opcode sequences and textCNN network. J. Inf. Secur. Appl. 2022, 67, 103151.