Contents

Contents 1
1 Results and discussion 2
1.1 Case study 2
1.2 Related works 3
1.3 Design of the Method 3
2 Performance metrics 3
2.1 False positive rate (FPR) 4

1 RESULTS AND DISCUSSION

This research developed the AGM-AB model that is implemented using the Python tool and for estimating the effectiveness of the introduced model, several metrics like accuracy, precision, FPR value, recall, F-measure, and AUC are calculated.

1.1 Case study

In this research, the novel deep learning approach is developed for increasing the security of VMs in the cloud service. Let 10 numbers of virtual machines in cloud service for analyzing the security.

$$F1 - score = 2 \int \frac{P * R}{P + R} dP$$

The F1-measure value of the introduced AGM-AB manner has been evaluated by prevailing methods like AMMD, IMI, TKRD, RF, AMD, and Ensemble learning.

The feature selection process is utilized to enhance the classification of benign and malware executable. Hence, the proposed model has effectively detected the normal executable files and malware. Malware classification using the proposed AGM-AB model is represented in Figure 1.

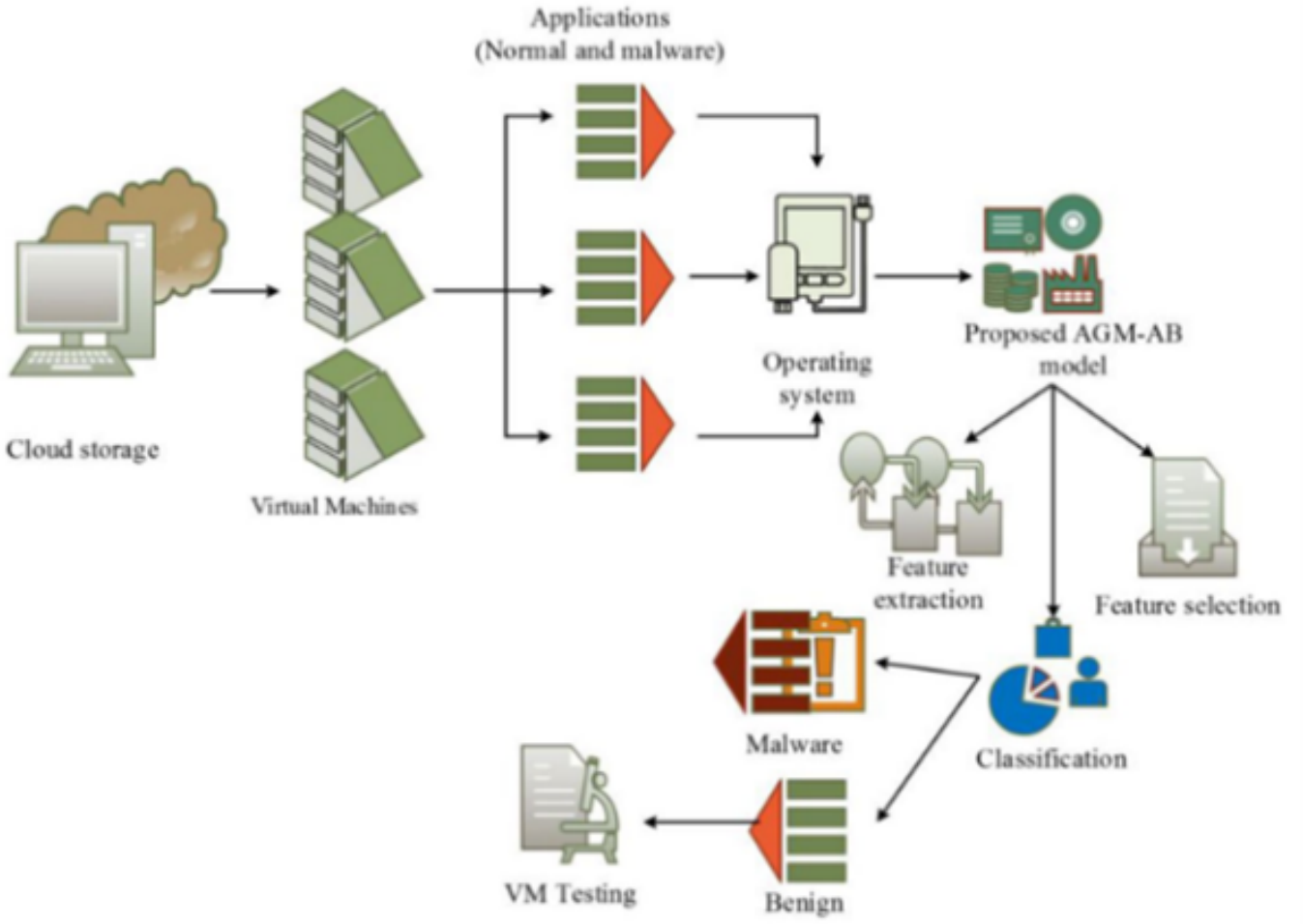

Fig. 1. Malware classification in VM

Table 1. Summary of state-of-the-art approaches

Author	Methods	Disadvantages
Mishra et al.	KVM Inspector and ML	Malicious users can easily access the security system.
Kumara and Jaidhar	ML, RF MFA, and leveraging VMI	Malware detection rate is limited.
Patil et al.	AMD	Malware dynamic behaviour analysis is required
Zhang et al.	Detection model using MFA	It has achieved a high error rate.
Kumara and Jaidhar	AMMDS	Processing time of the approach is high.
Kordestani et al.	LSS	FPR and error rate is high
Kordestani and Saif	observer-based mitigation and attack detection framework	High rate of vulnerable cyber attacks
Kiperberg	Virtualization based method	Less reliability
Proposed	AGM-AB	BUsing a hybrid system it will get a better outcome

1.2 Related works

Several existing techniques are reviewed above, from that some issues in VMI such as network error, malicious attacks, VMI security, high computation time and high FPR rate were recognized. These issues have been motivated this research to enhance the performance of detecting malware, True Positive Rate (TPR), False Prediction Rate (FPR), error rate and so on. The key contributions of this research work involves,

- Initially, the collected dataset is trained to the system that involves malware and benign files;
- Moreover, a novel AGM-AB algorithm is developed for detecting unknown malware functions from the benign program;
- Additionally, the African buffalo fitness module has been updated in the AGM manner to extract the features of the Virtual Machine Monitor (VMM);
- Here, the introduced AGM-AB model investigates the guest operating system, system calls, and kernel data for classifying the malware and benign files;
- Also, the AGM-AB approach is tested by launching faults and malware functions to demonstrate the effectiveness of the AGM-AB method;
- Subsequently, the implementation of this proposed AGM-AB approach is done in the Python tool and the metrics are computed;
- Finally, the proposed method is evaluated by prevailing approaches in terms of recall, accuracy, AUC, FPR, precision, and F-measure;

1.3 Design of the Method

This section details the design and operation of the proposed method to analyze malware in a Virtual Machine through Virtual Machine Introspection technique. It consists of the following five steps:

(1) Access to the asset: This is achieved through the interprocess mechanism called COM/XPCOM that implements the VirtualBox API;
(2) Collection: It generates a memory dump of the Virtual Machine volatile memory;
(3) Analysis: It translates the low-level bytes into high-level information with the help of the Volatility tool, through the profile of the virtual machine and extracts objects from the operating system;
(4) Logging: It generates the log of the malware analysis;
(5) Containment: The COM/XPCOM interprocess mechanism sends a killing command to finish the malware execution from outside the Virtual Machine;

2 PERFORMANCE METRICS

FN characterizes the false-negative rate for evaluating the inaccurate detection of the malware files and FP denotes the false-positive value for calculating the inaccurate detection of benign files (Table 1).

Initially, the dataset is collected from a GitHub source that is named as 'ember' for performing the process that involves malware and benign files. Initially, the dataset Ds is trained to the system using Eq:

$$D_s = \{ x_n y_n \} ; \{ n = 1, 2, 3 \ldots \}$$

Where, x_n is represents the benign executable files, y_n represents the malware files in the virtual machine

Table 2. Comparison of FPR

Number of executable files in VM	RF	AMD	Ensemble learning	TKRD	AMMD	IMI	AGM-AB
100	0.004	3.70	0.322	0.603	0.005	0.13	0.001
200	0.006	3.72	0.43	0.71	0.009	0.22	0.002.
300	0.027	3.85	0.58	0.88	0.0016	0.37	0.0035
400	0.093	3.95	0.67	1.02	0.0023	0.47	0.004
500	0.075	4.37	1.04	1.10	0.0034	0.53	0.005

It is calculated by the number of incorrect and correct predictions which is summarized by count values. It is used to summarize the performance of classified malware files on the position of tested and trained data. Additionally, the confusion matrix is represented in Figure 2, where, TP denotes the true positive for calculating the accurate detection of the malware files in the dataset.

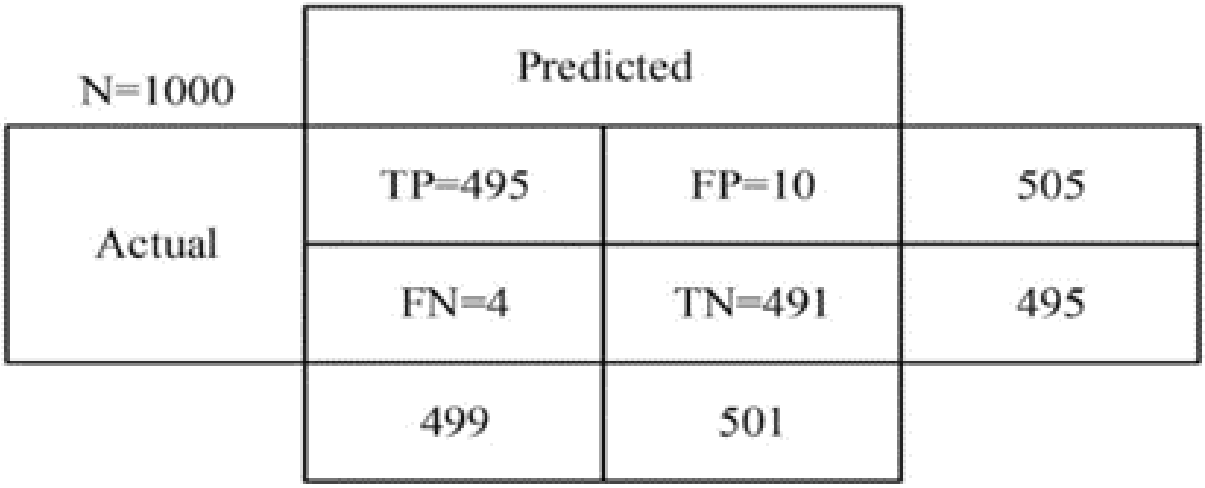

Fig. 2. Confusion matrix

2.1 False positive rate (FPR)

FPR denotes the false-positive ratio that is defined as the ratio between inaccurately detected benign files and the total quantity of FP and TN values, which is calculated using Eq (1),

$$A = \frac{\frac{TP+TN}{FPR+FN}}{TN + FP + TP + FN} \tag{1}$$

The FPR rate of the introduced AGM-AB approach is evaluated by prevailing methods like RF, AMD, TKRD, IMI, AMMD and Ensemble learning approaches, which are detailed in Table 2. The inaccurate detection of the executable files should be less for an efficient approach.