NSS
Neutrosophic Sets and Systems, Vol. 88, 2025
University of New Mexico

Interval Valued Neutrosophic Set with Machine Learning Model Dynamic Malware Detection in Digital Security

Mohanad Mousa Janat[1], Ahmed A El-Douh[2,5], Ahmed Abdelhafeez[3,5], Hanadi Ahmad Simmak[4]

[1]Medical Physics Department, College of Science, Ashur University, Baghdad, Iraq
[2]College of Informatics, Midocean University, 98123, Moroni, Comoros
[3]Computer Science Department, Faculty of Information Systems and Computer Science, October 6 University, Giza, 12585, Egypt
[4]Communication Engineering Department, Electrical and Electronic Engineering Faculty, Aleppo University, Syria
[5]Applied Science Research Center, Applied Science Private University, Amman

Abstract: Traditional signature-based detection techniques are useless against new forms of malware due to their fast development, which poses a serious cybersecurity risk. People, businesses, and governments are all affected by this expanding threat, highlighting the urgent need for robust malware detection systems. Due to their reliance on predetermined signatures, traditional machine learning-based techniques frequently fail to identify threats that have not yet been identified and instead rely on static and dynamic malware analysis. To improve malware detection performance across a variety of datasets, this study assesses traditional ML. Interval Valued Neutrosophic Set (IVNS) is used in this study to overcome vague information. The Neutrosophic Model is used to evaluate and rank six ML models. The results show support vector machine is the best ML Model for dynamic malware detection in digital security.

Keywords: Malware Classification and Detection; Security; Interval Valued Neutrosophic Set; Machine Learning Model.

1. Introduction

The Internet of Things (IoT) and its related applications have led to the development of the modern information society. However, security concerns significantly impede the full benefits of this economic revolution. Cybercriminals commonly target networks and individual computers with denial-of-service attacks and steal personal data for financial benefit.[1], [2]. These attackers use malicious software, or malware, to significantly endanger and compromise systems. Malware is computer software designed to harm an operating system (OS). Depending on its behavior and objective, malware can go by a variety of names, including adware, spyware, virus, worm, trojan, rootkit, backdoor, ransomware, and command and control C2 bot.

In cybersecurity, problems with malware identification and mitigation are continually evolving. As researchers develop new techniques, malware authors constantly improve their methods to evade detection[3]. In this study, we aimed to detect and classify memory-based obfuscated malware. I used the most recent dataset, the CIC-MalMem-2022, for my tests. In addition to identifying the presence of malware, this dataset provides details on its kind and family[4], [5]. Thus, we conducted two tests: one to determine the malware family (multi-class classification) and the other to determine the harmfulness of a specific sample (binary classification).

To do this, we have assessed several traditional machine learning (ML) methods, including support vector machines (SVM), logistic regression, decision trees, random forests, and k-nearest neighbor (KNN)[6], [7]. The open, adaptable, and portable nature of wireless communication makes security risks worse. Intrusion detection systems (IDS), both host and network, are essential for protecting these networks against these threats.[8]. An efficient, dependable, and resilient intrusion detection system (IDS) must be able to manage alarm frequency, minimize false positives, and identify threats. Machine learning is being used in recent studies to improve IDS capabilities. ML models are applied in different applications.[9], [10].

The ability of machine learning algorithms to identify patterns in large datasets is essential for identifying security issues. Conventional IDS makes use of several ML methods, including decision trees, Support Vector Machines (SVM), and k-nearest neighbors. The methods used by malware producers to avoid detection change as the industry does[11], [12]. To detect and categorize memory-obfuscated malware, we used the most current CIC-MalMem-2022 dataset.

Neutrosophic hypotheses were put out by Smarandache. [13], [14]. A new area of philosophy known as "neutrosophy" was presented. It addresses the nature, breadth, and genesis of neutralities as well as how they interact with other ideational spectra. The neutrosophic set (NS), neutrosophic logic, neutrosophic probability, neutrosophic statistics, and so on are all based on neutrosophy. According to NS theory, each event has a degree of truth as well as a degree of falsehood and indeterminacy that must be considered separately.[15].

The main contributions of this study are:

1. Machine learning models are used for dynamic malware detection in digital security for intrusion detection.
2. Six machine learning models are used in this study to obtain higher accuracy and performance from the dataset.
3. Large-scale datasets are used in this study.
4. The interval valued neutrosophic set framework is used to evaluate six ML models and select the best one based on four evaluation matrices.

2. Interval Valued Neutrosophic Set

This section shows the definition of an interval valued neutrosophic set (IVNS)[16], [17]. We show some operations of IVNS, such as.

$$HD = \begin{pmatrix} \left[1-\left(1-T_D^L(N)\right)^H, 1-\left(1-T_D^U(N)\right)^H\right], \\ \left[\left(I_D^L(N)\right)^H, \left(I_D^U(N)\right)^H\right], \\ \left[\left(F_D^L(N)\right)^H, \left(F_D^U(N)\right)^H\right] \end{pmatrix} \quad (1)$$

$$D^H = \begin{pmatrix} \left[\left(T_D^L(N)\right)^H, \left(T_D^U(N)\right)^H\right], \\ \left[1-\left(1-I_D^L(N)\right)^H, 1-\left(1-I_D^U(N)\right)^H\right], \\ \left[1-\left(1-F_D^L(N)\right)^H, 1-\left(1-F_D^U(N)\right)^H\right] \end{pmatrix} \quad (2)$$

$$D+Z = \begin{pmatrix} [T_D^L(N)+T_Z^L(N)-T_D^L(N)T_Z^L(N), T_D^U(N)+T_Z^U(N)-T_D^U(N)T_Z^U(N)], \\ [I_D^L(N)I_Z^L(N), I_D^U(N)I_Z^U(N)], \\ [F_D^L(N)F_Z^L(N), F_D^U(N)F_Z^U(N)] \end{pmatrix} \quad (3)$$

$$DZ = \begin{pmatrix} [T_D^L(N)T_Z^L(N), T_D^U(N)T_Z^U(N)], \\ [I_D^L(N)+I_Z^L(N)-I_D^L(N)I_Z^L(N), I_D^U(N)+I_Z^U(N)-I_D^U(N)I_Z^U(N)], \\ [F_D^L(N)+F_Z^L(N)-F_D^L(N)F_Z^L(N), F_D^U(N)+F_Z^U(N)-F_D^U(N)T_Z^U(N)] \end{pmatrix} \quad (4)$$

$$D-Z = \begin{pmatrix} [T_D^L(N)-T_Z^U(N), T_D^U(N)-T_Z^L(N)], \\ \left[\max\left(I_D^L(N), I_Z^L(N)\right), \max\left(I_D^U(N), I_Z^U(N)\right)\right], \\ [F_D^L(N)-F_Z^U(N), F_D^U(N)-F_Z^L(N)] \end{pmatrix} \quad (5)$$

$$\frac{D}{H} = \begin{pmatrix} \left[\min\left(\frac{T_D^L(N)}{H}, 1\right), \min\left(\frac{T_D^U(N)}{H}, 1\right)\right], \\ \left[\min\left(\frac{I_D^L(N)}{H}, 1\right), \min\left(\frac{I_D^U(N)}{N}, 1\right)\right] \\ \left[\min\left(\frac{F_D^L(N)}{H}, 1\right), \min\left(\frac{F_D^U(N)}{H}, 1\right)\right] \end{pmatrix} \quad (6)$$

$$D^{-1} = \begin{pmatrix} \left[\left(T_D^L(N)\right)^{-1}, \left(T_D^U(N)\right)^{-1}\right], \\ \left[\left(I_D^L(N)\right)^{-1}, \left(I_D^U(N)\right)^{-1}\right], \\ \left[\left(F_D^L(N)\right)^{-1}, \left(F_D^U(N)\right)^{-1}\right] \end{pmatrix} \quad (7)$$

3. Results

Using a current dataset is essential for this research. It is helpful for equitably assessing novel techniques and figuring out how well they work in practical situations. We utilized the CIC-MalMem-2022 dataset in this experiment. Examples of both obfuscated and non-obfuscated malware are included in its collection. The study includes common malware types like trojans, ransomware, and spyware to make it more realistic.

With 58,596 records that are evenly split between 50% malicious and 50% benign memory dumps, the CIC-MalMem-2022 dataset replicates real-world situations. It comprises several malware families, including Trojan Horse, Spyware, and Ransomware. Memory dump analysis is the basis for feature selection to identify malware that has been disguised. Making sure that both dangerous and benign samples are distributed equally helps to combat data imbalance. Figure 1 shows the four class counts. Figure 2 shows the rate of four classes in the dataset. Figure 3 shows the counts of each category. Figure 4 shows the correlation matrix.

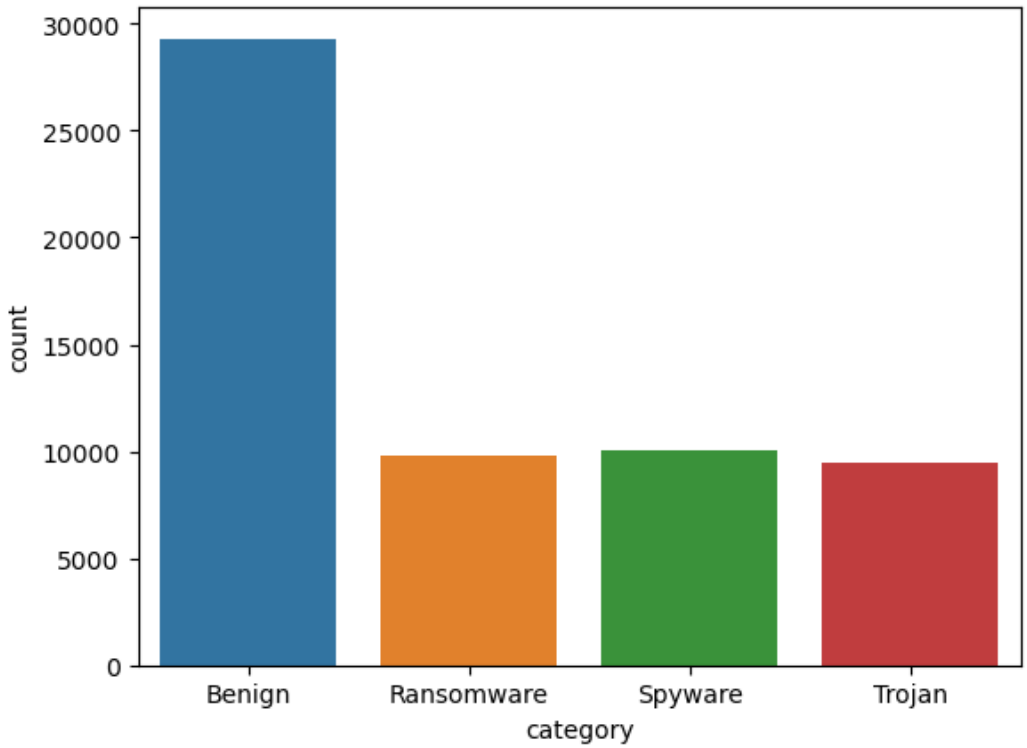

Figure 1. Four class values.

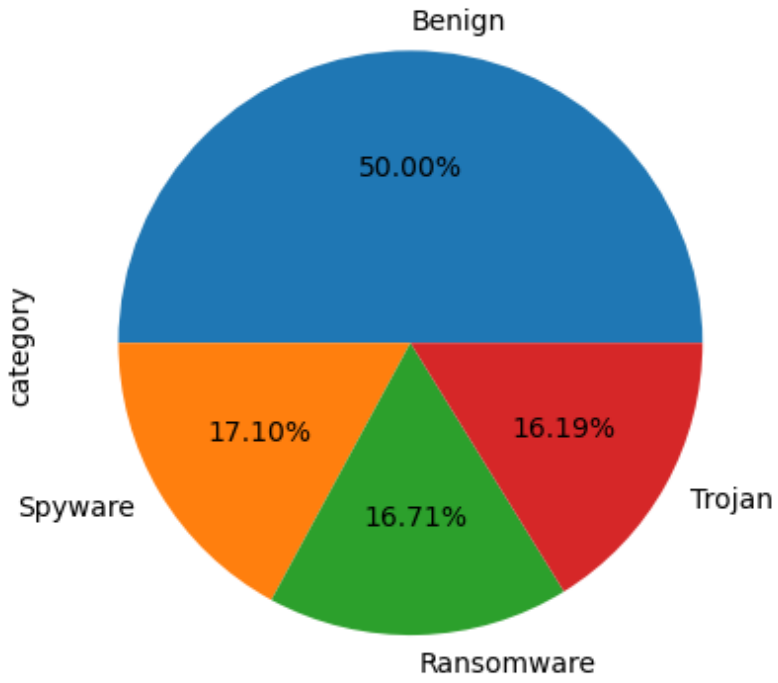

Figure 2. The rate of four classes.

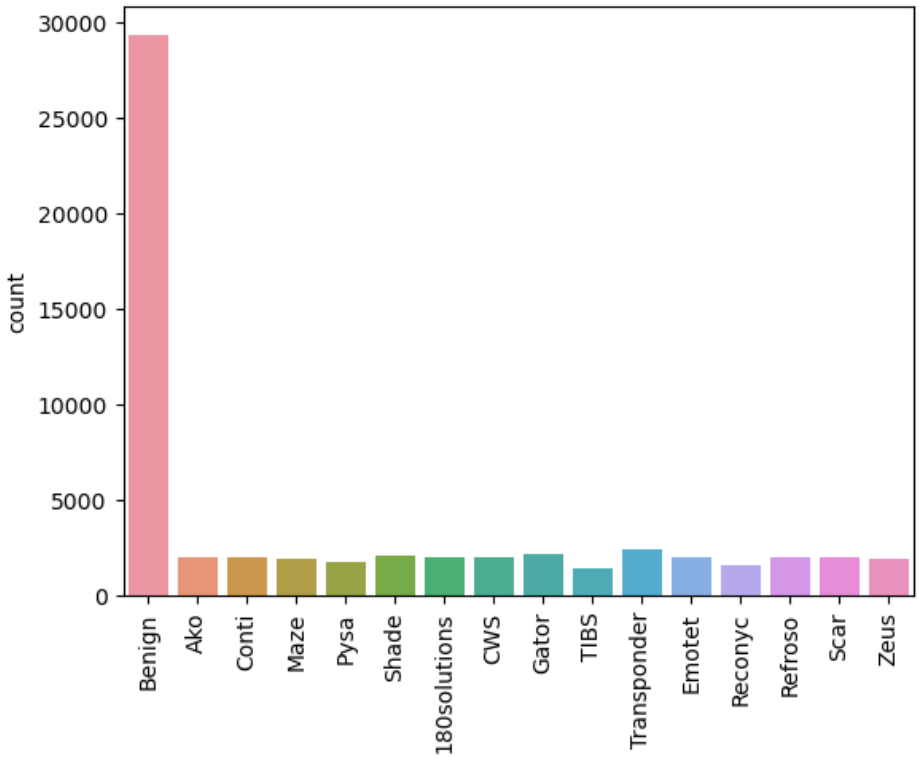

Figure 3. Counts of country classes.

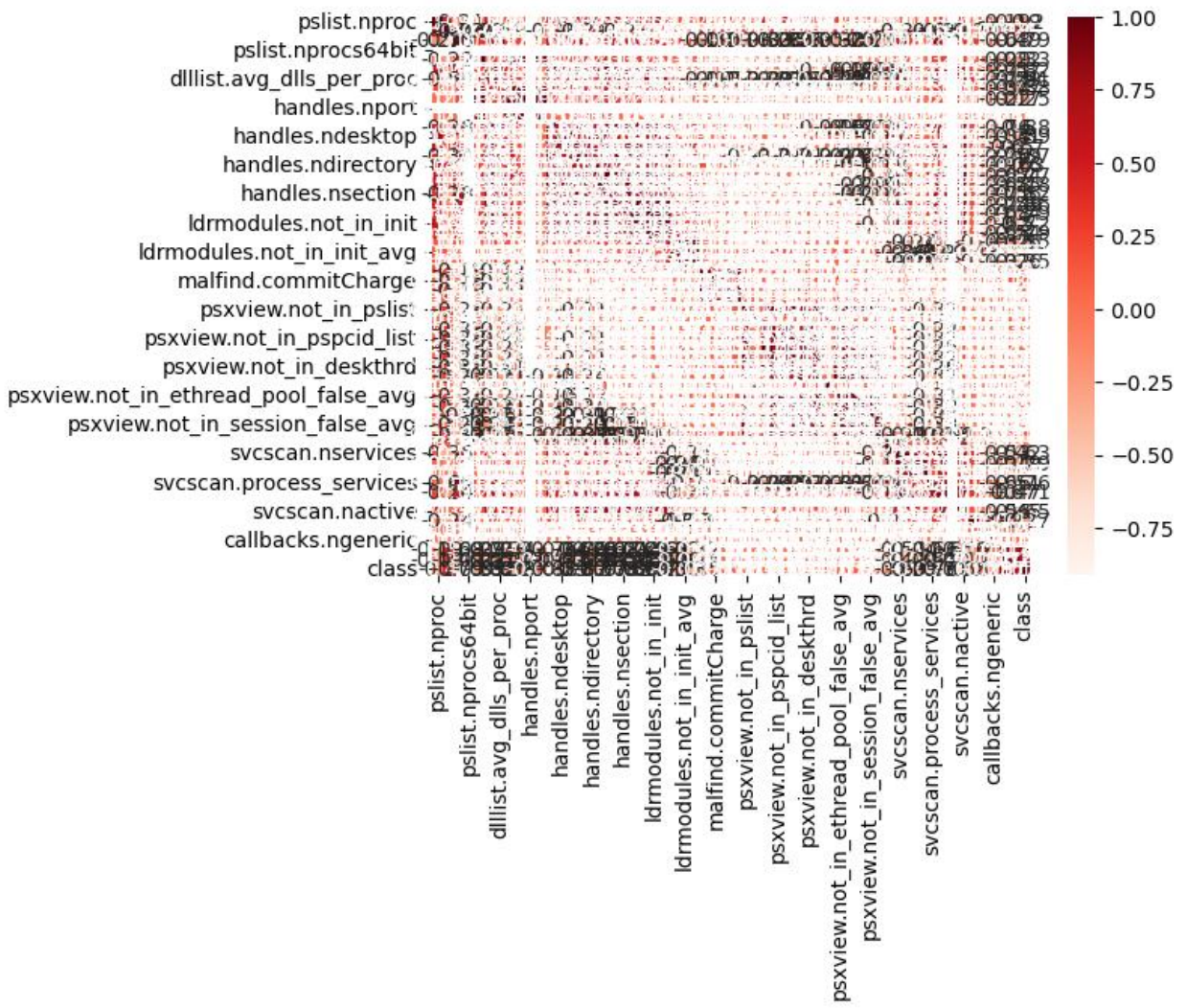

Figure 4. The correlation matrix.

Then we applied six ML models, including logistic regression (LR), k-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), random forest (RF), and XGBoost, as shown in Table 1.

Table 1. Results of six ML models.

	Accuracy	Precision	Recall	F1-score
KNN	0.999886	0.999887	1	0.999773
SVM	1	1	1	1
DT	1	1	1	1
XGBoost	1	1	1	1
LR	0.999886	0.999887	1	0.999773
RF	1	1	1	1

Then we show the neutrosophic set model to select the best ML model. Six experts created the decision matrix as shown in Table 2. Then we compute the criteria weights using the average method as: 0.241354409, 0.252587659, 0.249378159, 0.256679772.

Table 2. Decision matrix.

	$MDSC_1$	$MDSC_2$	$MDSC_3$	$MDSC_4$
$MDSA_1$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])
$MDSA_2$	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.5,0.5],[0.6,0.7],[0.4,0.5])
$MDSA_3$	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.5,0.6],[0.5,0.6],[0.4,0.5])
$MDSA_4$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.6,0.7],[0.4,0.5],[0.3,0.4])	([0.6,0.7],[0.4,0.5],[0.3,0.4])
$MDSA_5$	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.1,0.2],[0.1,0.2],[0.8,0.9])
$MDSA_6$	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.2,0.3],[0.3,0.4],[0.7,0.8])
	$MDSC_1$	$MDSC_2$	$MDSC_3$	$MDSC_4$
$MDSA_1$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])
$MDSA_2$	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.5,0.5],[0.6,0.7],[0.4,0.5])
$MDSA_3$	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.5,0.6],[0.5,0.6],[0.4,0.5])
$MDSA_4$	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.6,0.7],[0.4,0.5],[0.3,0.4])
$MDSA_5$	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.1,0.2], [0.1,0.2],[0.8,0.9])
$MDSA_6$	([0.5,0.5], [0.6,0.], [[0.4,0.5])	([0.4,0.5], [0.5,0.6],[0.5,0.6])	([0.3,0.4], [0.4,0.], [[0.6,0.7])	([0.2,0.3], [0.3,0.], [[0.7,0.8])
	$MDSC_1$	$MDSC_2$	$MDSC_3$	$MDSC_4$
$MDSA_1$	([0.1,0.2], [0.1,0.2],[0.8,0.9])	([0.2,0.3], [0.3,0.4],[0.7,0.8])	([0.3,0.4], [0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])
$MDSA_2$	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.5,0.5],[0.6,0.7],[0.4,0.5])
$MDSA_3$	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.5,0.6],[0.5,0.6],[0.4,0.5])
$MDSA_4$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.1,0.2],[0.1,0.2],[0.8,0.9])
$MDSA_5$	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])
$MDSA_6$	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.2,0.3],[0.3,0.4],[0.7,0.8])
	$MDSC_1$	$MDSC_2$	$MDSC_3$	$MDSC_4$
$MDSA_1$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.1,0.2],[0.1,0.2],[0.8,0.9])
$MDSA_2$	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.5,0.6],[0.5,0.6],[0.4,0.5])
$MDSA_3$	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.1,0.2],[0.1,0.2],[0.8,0.9])
$MDSA_4$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.5,0.6],[0.5,0.6],[0.4,0.5])
$MDSA_5$	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.5,0.5],[0.6,0.7],[0.4,0.5])
$MDSA_6$	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.2,0.3],[0.3,0.4],[0.7,0.8])
	$MDSC_1$	$MDSC_2$	$MDSC_3$	$MDSC_4$
$MDSA_1$	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])
$MDSA_2$	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.4,0.5],[0.5,0.6],[0.5,0.6])
$MDSA_3$	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])
$MDSA_4$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.2,0.3],[0.3,0.4],[0.7,0.8])
$MDSA_5$	([0.5,0.6],[0.5,0.6],[0.4,0.5])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.1,0.2],[0.1,0.2],[0.8,0.9])
$MDSA_6$	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.2,0.3],[0.3,0.4],[0.7,0.8])
	$MDSC_1$	$MDSC_2$	$MDSC_3$	$MDSC_4$
$MDSA_1$	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])
$MDSA_2$	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])
$MDSA_3$	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.4,0.5],[0.5,0.6],[0.5,0.6])
$MDSA_4$	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.3,0.4],[0.4,0.5],[0.6,0.7])
$MDSA_5$	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.2,0.3],[0.3,0.4],[0.7,0.8])	([0.1,0.2],[0.1,0.2],[0.8,0.9])	([0.2,0.3],[0.3,0.4],[0.7,0.8])
$MDSA_6$	([0.5,0.5],[0.6,0.7],[0.4,0.5])	([0.4,0.5],[0.5,0.6],[0.5,0.6])	([0.3,0.4],[0.4,0.5],[0.6,0.7])	([0.2,0.3],[0.3,0.4],[0.7,0.8])

This paper multiplies the weights of the criteria by the decision matrix as shown in Table 3. The sum of each alternative is shown in Figure 5. The ranks of alternatives are shown in Figure 5. The results show that SVM is the best.

Table 3. Weighted decision matrix.

	$MDSC_1$	$MDSC_2$	$MDSC_3$	$MDSC_4$
$MDSA_1$	0.067378	0.090932	0.107233	0.117003
$MDSA_2$	0.103782	0.115138	0.116169	0.131976
$MDSA_3$	0.098151	0.109244	0.119494	0.126201
$MDSA_4$	0.076027	0.113454	0.126767	0.130479
$MDSA_5$	0.134153	0.10756	0.068995	0.086202
$MDSA_6$	0.125504	0.126294	0.107233	0.092405

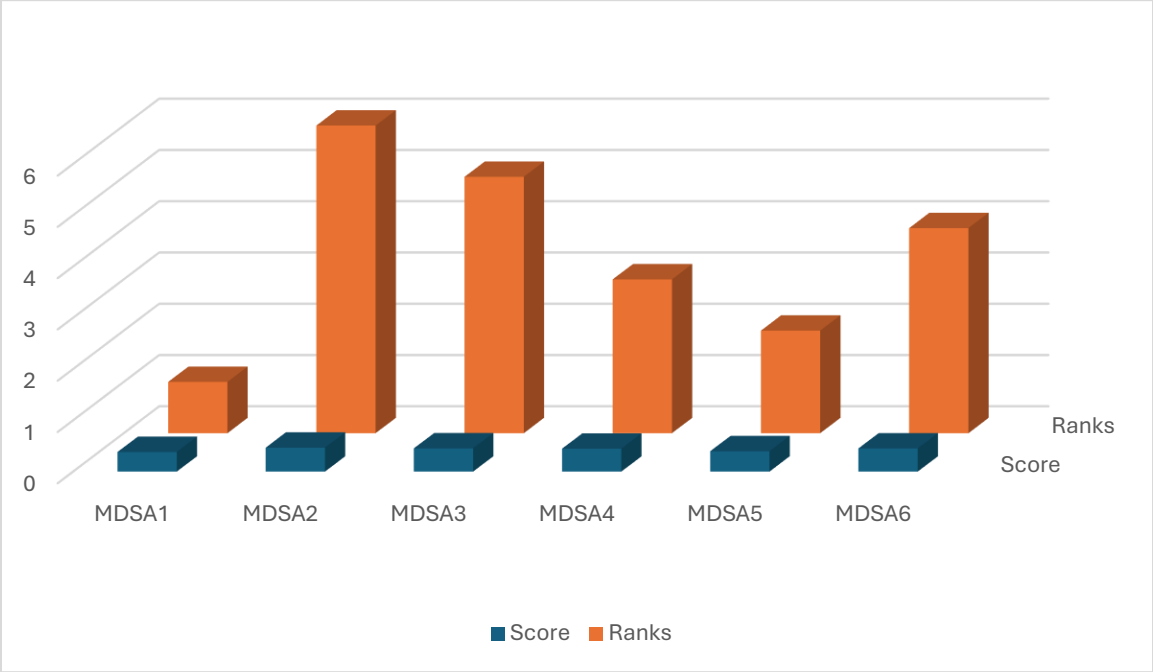

Figure 5. Ranks of alternatives.

4. Conclusions

This study effectively used machine learning techniques to create a dynamic malware detection system, notably using SVM, RF, XGBoost, DT, KNN, and LR models. With an accuracy of 100.00%, the system showed notable gains in resilience and accuracy in malware identification. The model was able to access a variety of data sources thanks to the dual dataset method, which improved its capacity to identify and categorize various kinds of malware more accurately. The attained accuracy demonstrates the revolutionary potential of machine learning in cybersecurity, as these models provide a dependable defense against sophisticated attacks by efficiently analyzing intricate patterns and abnormalities within malware. These methods provide strong protection against new threats by improving detection skills and adjusting to the constantly changing cybersecurity landscape. Interval Valued Neutrosophic Set (IVNS) to deal with

uncertainty and vague information. The neutrosophic sets are used to evaluate six ML models. The results show that the SVM is the best model.

References

[1] M. Oppal, S. Whitmore, V. Holloway, and W. Abernathy, "Deep ransomware detection through dynamic vulnerability profiling for real-time threat identification," 2024.

[2] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—A state of the art survey," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–48, 2019.

[3] T. Alzahrani, "Advanced Techniques for Dynamic Malware Detection and Classification in Digital Security Using Deep Learning," *Comput. Mater. Contin.*, vol. 83, no. 3, 2025.

[4] P. Yan and Z. Yan, "A survey on dynamic mobile malware detection," *Softw. Qual. J.*, vol. 26, no. 3, pp. 891–919, 2018.

[5] N. Usman *et al.*, "Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics," *Futur. Gener. Comput. Syst.*, vol. 118, pp. 124–141, 2021.

[6] R. Karim and M. I. Asjad, "Improving Equity in Healthcare: Machine Learning-Based Thyroid Disease Classification," *SciNexuses*, vol. 1, pp. 139–147, 2024.

[7] A. H. Abed and M. A. Ebrahim, "Blood Cancer Detection from Blood Smear Images using Machine Learning Algorithms," *SciNexuses*, vol. 1, pp. 160–173, 2024.

[8] P. V. Shijo and A. Salim, "Integrated static and dynamic analysis for malware detection," *Procedia Comput. Sci.*, vol. 46, pp. 804–811, 2015.

[9] A. S. Aziz, K. Ibrahim, A. Elsharkawy, and N. Khaliel, "Anticipating Diabetes using Fusion-Ensemble Machine Learning Techniques," *SciNexuses*, vol. 1, pp. 44–51, 2024.

[10] A. M. Ali and S. A. Esmail, "Prediction of COVID-19 Patients using Machine Learning Algorithms," *SciNexuses*, vol. 1, pp. 58–69, 2024.

[11] T. Bhatia and R. Kaushal, "Malware detection in Android based on dynamic analysis," in *2017 International conference on cyber security and protection of digital services (Cyber security)*, IEEE, 2017, pp. 1–6.

[12] R. S. Mihalache, D. T. Gavrilut, and A. D. Gabriel, "Real-Time Deep Learning-Based Malware Detection Using Static and Dynamic Features," in *ICAART (3)*, 2024, pp. 226–234.

[13] F. Smarandache, *A unifying field in logics: neutrosophic logic. Neutrosophy, neutrosophic set, neutrosophic probability: neutrosophic logic. Neutrosophy, neutrosophic set, neutrosophic probability*. Infinite Study, 2005.

[14] F. Smarandache, *Introduction to neutrosophic measure, neutrosophic integral, and neutrosophic probability*. Infinite Study, 2013.

[15] H. Zhang, J. Wang, and X. Chen, "An outranking approach for multi-criteria decision-making problems with interval-valued neutrosophic sets," *Neural Comput. Appl.*, vol. 27, pp. 615–627, 2016.

[16] E. Bolturk and C. Kahraman, "A novel interval-valued neutrosophic AHP with cosine similarity measure," *Soft Comput.*, vol. 22, no. 15, pp. 4941–4958, 2018.

[17] I. Deli, "Interval-valued neutrosophic soft sets and their decision making," *Int. J. Mach. Learn. Cybern.*, vol. 8, pp. 665–676, 2017.

Received: Dec. 22, 2024. Accepted: June 30, 2025