NSS
Neutrosophic Sets and Systems, Vol. 85, 2025
University of New Mexico

Windows Malware Detection Under the Machine Learning Models and Neutrosophic Numbers

Alber S. Aziz1. Mohamed eassa2, Ahmed Abdelhafeez3, Ahmed A. Metwaly4, Ashraf. M. Hussein5 and, Nariman A. Khalil6

1,2,3 Computer Science Department, Faculty of Information Systems and Computer Science, October 6th University, Giza, 12585, Egypt Albershawky.csis@o6u.edu.eg; mohamed.eassa.cs@o6u.edu.eg; aahafeez.scis@o6u.edu.eg

2,3 Applied Science Research Center. Applied Science Private University, Amman, Jordan

4 Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt, a.metwaly23@fci.zu.edu.eg

5 Department of computer science, Faculty of computer and artificial intelligence, Modern university for information and technology, Cairo, Egypt, Ashraf-abdelaliem@adj.aast.edu

6 Assistant professor, Egyptian Chinese College for Applied Technology (ECCAT), Suez Canal University, Narimankhaliel.eccat@suez.edu.eg

Abstract:

Significant cybersecurity risks are posed by malware assaults on Windows computers, which call for efficient detection and prevention systems. Supervised machine learning classifiers have shown great promise in the field of malware detection. Comprehensive research comparing the effectiveness of various classifiers, particularly for Windows malware detection, is still required. Closing this gap can yield valuable information for improving cybersecurity tactics. A thorough comparison of supervised classifiers for Windows malware detection is lacking, even though several research have investigated malware detection using machine learning approaches. Determining the relative efficacy of these classifiers can help choose the best detection techniques and enhance security protocols in general. This study applies to 6 ML models for Windows malware detection. After that, we evaluate these models using the neutrosophic set to overcome the uncertainty information. The single values neutrosophic sets (SVNSs) are used in this study. The EDAS method is used to select the based model under the evaluation matrices.

Keywords: Single Valued Neutrosophic Sets; Machine Learning; EDAS; Windows Malware Detection.

1. Introduction

The growing danger of malware in modern digital contexts, particularly in Windows operating systems, emphasizes how urgently strong detection systems are needed. Malicious software, which can range from viruses to ransomware, presents serious hazards such as compromised systems, interruptions to operations, and data breaches.[1], [2]. As a result, creating efficient malware detection techniques has become crucial to protecting systems and data integrity.

Using labeled datasets to train classifiers that can recognize dangerous patterns and behaviors, supervised machine learning presents a potential approach to malware detection. ML models have become prominent candidates for malware identification among the wide range of supervised learning techniques.[3], [4]. Nevertheless, the literature currently in publication noticeably lacks a thorough comparison of these classifiers that are especially suited for Windows malware detection.

Like any other software industry, the malware sector is well-funded, well-organized, and steady. It is also taking steps to circumvent conventional security measures. Microsoft chose to implement countermeasures to identify potential attacks before they occurred and then strengthen and protect its system to address the problem of malware assaults on Windows computers.[5], [6]. This is a crucial precaution to take because, should the malware successfully infiltrate the system and gain control, sensitive data or the end user's or company's valuable information be compromised, potentially leading to a sharp decline in the clients' confidence in Microsoft's system.[7], [8].

Therefore, Microsoft challenged data scientists and analysts worldwide to anticipate using their data, which is the actual data, but concealing the confidential information of end users. Many data-driven methods are used in research to identify the likelihood of malware attacks on computers in advance so that they can be effectively prevented and the related losses reduced.[9], [10]. While some of these methods target executable processes, others extract patterns from malware data and compare them to programs to determine if they are malware.

Not all attribute values can be precisely quantified due to life's uncertainties, particularly when it comes to qualitative markers. We frequently represent this ambiguity using linguistic variables or imprecise figures. In 1965, Zadeh originally proposed fuzzy sets (FSs) to express uncertain data. Its main goal is to illustrate the ambiguity and uncertainty present in the actual world by using the membership function. Artificial intelligence has made extensive use of FS theory ever since.[11], [12].

In 1986, Atanassov introduced intuitionistic fuzzy sets (IFSs), which are more universally applicable than FSs, to further improve FSs. More possibilities for decision-making were then made possible by the emergence of interval fuzzy sets, interval intuitionistic fuzzy sets, hesitation fuzzy sets, and other extended sets. There are still restrictions, though, such as the inability to deal with inconsistent and discontinuous information. The single-valued neutrosophic set (SVNSs), which is defined by H. Wang et al. as an extension of FSs and IFSs, uses three decimals between 0 and 1 to indicate the degree of truth, uncertainty, and indeterminacy of information, respectively. The SVNSs have been extensively researched by academics because of their exceptional qualities, such as flexibility and comprehensiveness.[13], [14].

2. Methodology

This section shows the definitions of the single-valued neutrosophic set (SVNS), the steps of the EDAS method to rank the alternatives, and the steps of the ML models.

We show the operations of the SVNS. Let two single-valued neutrosophic numbers (SVNNs) such as[15], [16]:

$X_1 = t_{X_1}(D), i_{X_1}(D), f_{X_1}(D)$ and $X_2 = t_{X_2}(D), i_{X_3}(D), f_{X_4}(D)$

$$X_1^c = \left(f_{X_1}(D), 1 - i_{X_1}(D), t_{X_1}(D)\right)$$

$$X_1 \cup X_2 = \begin{pmatrix} \max\{t_{X_1}(D), t_{X_2}(D)\}, \\ \min\{i_{X_1}(D), i_{X_2}(D)\}, \\ \min\{f_{X_1}(D), f_{X_2}(D)\} \end{pmatrix} \quad (1)$$

$$X_1 \cap X_2 = \begin{pmatrix} \min\{t_{X_1}(D), t_{X_2}(D)\}, \\ \max\{i_{X_1}(D), i_{X_2}(D)\}, \\ \max\{f_{X_1}(D), f_{X_2}(D)\} \end{pmatrix} \quad (2)$$

$$X_1 + X_2 = \begin{pmatrix} t_{X_1}(D) + t_{X_2}(D) - t_{X_1}(D)t_{X_2}(D), \\ i_{X_1}(D)i_{X_2}(D), \\ f_{X_1}(D)f_{X_2}(D) \end{pmatrix} \quad (3)$$

$$X_1X_2 = \begin{pmatrix} t_{X_1}(D)t_{X_2}(D), \\ i_{X_1}(D) + i_{X_2}(D) - i_{X_1}(D)i_{X_2}(D), \\ f_{X_1}(D) + f_{X_2}(D) - f_{X_1}(D)f_{X_2}(D) \end{pmatrix} \quad (4)$$

$$hX_1 = \begin{pmatrix} 1 - \left(1 - t_{X_1}(D)\right)^h, \\ \left(i_{X_1}(D)\right)^h, \\ \left(f_{X_1}(D)\right)^h \end{pmatrix} \quad (5)$$

$$X_1^h = \begin{pmatrix} \left(t_{X_1}(D)\right)^h, \\ 1-\left(1-i_{X_1}(D)\right)^h, \\ 1-\left(1-f_{X_1}(D)\right)^h \end{pmatrix} \quad (6)$$

The steps of the EDAS method are shown such as:

Determine the average values such as:

$$A_j = \frac{\sum_{i=1}^{m} x_{ij}}{m} \quad (7)$$

Determine the positive and negative distance from the A_j

$$Q_{ij} = \frac{\max\left(0, \left(x_{ij} - A_j\right)\right)}{A_j} \quad (8)$$

$$U_{ij} = \frac{\max\left(0, \left(A_j - x_{ij}\right)\right)}{A_j} \quad (9)$$

$$Q_{ij} = \frac{\max\left(0, \left(A_j - x_{ij}\right)\right)}{A_j} \quad (10)$$

$$U_{ij} = \frac{\max\left(0, \left(x_{ij} - A_j\right)\right)}{A_j} \quad (11)$$

Obtain the weighted Q_{ij} and U_{ij}

$$SQ_i = \sum_{j=1}^{n} Q_{ij} w_j \quad (12)$$

$$SU_i = \sum_{j=1}^{n} U_{ij} w_j \quad (13)$$

Obtain the weighted normalized. Q_{ij} and U_{ij}

$$NSQ_i = \frac{SQ_i}{\max(SQ)} \quad (14)$$

$$NSU_i = \frac{SU_i}{\max(SU_i)} \quad (15)$$

Obtain the appraisal value.

$$H_i = 0.5 * (NSQ_i + NSU_i) \quad (16)$$

Cleaning the data is the first stage, which we covered in depth in the last section. Next, we divided the data into two halves. Data for testing and training. Models are trained using training data, and the accuracy of the projected outcomes is then assessed using testing data. Training Using our data and the method, we train the classifier in this stage. To forecast the target class based on patterns learned during a training phase, the classifier is given the unseen data.

The accuracy of the model is then assessed by comparing the predicted class with the actual target class. We have used several of them for this study and examined the outcomes of each method. Decision Tree, random forest, XGBoost, AdaBoosting, gradient boosting, and stochastic gradient descent models are employed.

3. Implementation

This section shows the results of the ML models under the different metrics as shown in Table 1. We implemented the ML models on the Windows malware dataset. We show the XGBoost model has higher accuracy. Figure 1 shows the ROC-Curve of the decision tree model.

Table 1. ML models.

	Accuracy	Precision	Recall	F1-score
Decision Tree	0.9334	0.933413	0.933413	0.933413
Random Forest	0.9466	0.946551	0.946551	0.946551
XGBoost	0.9486	0.948641	0.948641	0.948641
Gradient Boost	0.921469	0.921469	0.921469	0.921469
AdaBoost	0.850105	0.850105	0.850105	0.850105
stochastic gradient descent model	0.848313	0.848313	0.848313	0.848313

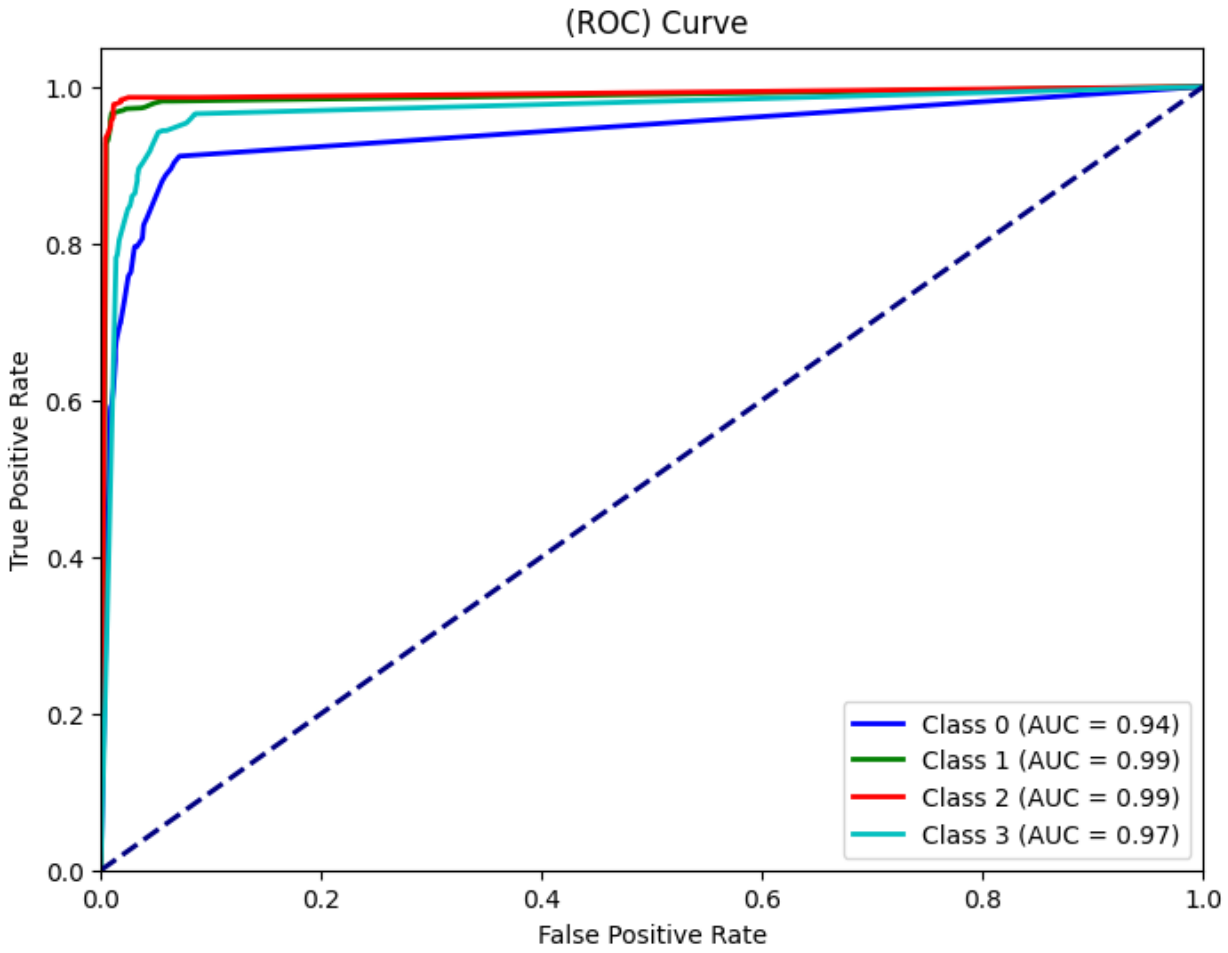

Figure 1. The ROC-Curve of the decision tree.

4. Neutrosophic Results

This section shows the results of the neutrosophic model selecting the best ML model under different matrices. Three experts evaluated the criteria and alternatives as shown in Table 2. The criteria weights are computed as: W1=0.208025343, W2=0.268215417, W3=0.268215417, W4=0.255543823.

Table 2. The decision matrix.

	WMC_1	WMC_2	WMC_3	WMC_4
WMA_1	(0.2,0.7,0.8)	(0.8,0.2,0.3)	(0.7,0.3,0.4)	(0.6,0.4,0.5)
WMA_2	(0.1,0.8,0.9)	(0.2,0.7,0.8)	(0.4,0.5,0.6)	(0.4,0.5,0.6)
WMA_3	(0.3,0.6,0.7)	(0.1,0.8,0.9)	(0.2,0.7,0.8)	(0.3,0.6,0.7)
WMA_4	(0.2,0.7,0.8)	(0.4,0.5,0.6)	(0.1,0.8,0.9)	(0.2,0.7,0.8)
WMA_5	(0.9,0.1,0.2)	(0.3,0.6,0.7)	(0.2,0.7,0.8)	(0.1,0.8,0.9)
WMA_6	(0.5,0.5,0.5)	(0.2,0.7,0.8)	(0.9,0.1,0.2)	(0.6,0.4,0.5)
	WMC_1	WMC_2	WMC_3	WMC_4

WMA_1	(0.4,0.5,0.6)	(0.8,0.2,0.3)	(0.7,0.3,0.4)	(0.6,0.4,0.5)
WMA_2	(0.7,0.3,0.4)	(0.2,0.7,0.8)	(0.4,0.5,0.6)	(0.2,0.7,0.8)
WMA_3	(0.2,0.7,0.8)	(0.1,0.8,0.9)	(0.2,0.7,0.8)	(0.1,0.8,0.9)
WMA_4	(0.1,0.8,0.9)	(0.6,0.4,0.5)	(0.1,0.8,0.9)	(0.7,0.3,0.4)
WMA_5	(0.4,0.5,0.6)	(0.9,0.1,0.2)	(0.6,0.4,0.5)	(0.6,0.4,0.5)
WMA_6	(0.5,0.5,0.5)	(0.8,0.2,0.3)	(0.9,0.1,0.2)	(0.9,0.1,0.2)
	WMC_1	WMC_2	WMC_3	WMC_4
WMA_1	(0.2,0.7,0.8)	(0.8,0.2,0.3)	(0.7,0.3,0.4)	(0.6,0.4,0.5)
WMA_2	(0.1,0.8,0.9)	(0.2,0.7,0.8)	(0.4,0.5,0.6)	(0.5,0.5,0.5)
WMA_3	(0.2,0.7,0.8)	(0.1,0.8,0.9)	(0.9,0.1,0.2)	(0.8,0.2,0.3)
WMA_4	(0.3,0.6,0.7)	(0.2,0.7,0.8)	(0.2,0.7,0.8)	(0.2,0.7,0.8)
WMA_5	(0.4,0.5,0.6)	(0.9,0.1,0.2)	(0.1,0.8,0.9)	(0.1,0.8,0.9)
WMA_6	(0.5,0.5,0.5)	(0.8,0.2,0.3)	(0.7,0.3,0.4)	(0.6,0.4,0.5)

Determine the average values using eq. (7).

Determine the positive and negative distance from the A_jUsing equations. (8-11) as shown in Table 3.

Obtain the weighted Q_{ij} and U_{ij} Using equations. (12-13) as shown in Table 4.

Obtain the weighted normalized. Q_{ij} and U_{ij} Using equations. (14-15).

Obtain the appraisal value using eq. (16) as shown in Table 5. We rank the alternatives. We show the XGBoost model is the best model.

Table 3. The values of distance.

	WMC_1	WMC_2	WMC_3	WMC_4
WMA_1	0	0.629921	0.417323	0.264463
WMA_2	0	0	0	0
WMA_3	0	0	0	0
WMA_4	0	0	0	0
WMA_5	0.583756	0.464567	0	0
WMA_6	0.370558	0.251969	0.700787	0.487603
	WMC_1	WMC_2	WMC_3	WMC_4
WMA_1	0.177665	0	0	0
WMA_2	0.147208	0.503937	0.07874	0.132231
WMA_3	0.269036	0.716535	0.055118	0.082645
WMA_4	0.360406	0.125984	0.645669	0.157025
WMA_5	0	0	0.338583	0.380165
WMA_6	0	0	0	0

Table 4. The values of weighted Q_{ij} and U_{ij}.

	WMC_1	WMC_2	WMC_3	WMC_4
WMA_1	0	0.168955	0.111932	0.067582
WMA_2	0	0	0	0
WMA_3	0	0	0	0
WMA_4	0	0	0	0
WMA_5	0.121436	0.124604	0	0
WMA_6	0.077086	0.067582	0.187962	0.124604
	WMC_1	WMC_2	WMC_3	WMC_4
WMA_1	0.036959	0	0	0
WMA_2	0.030623	0.135164	0.021119	0.033791
WMA_3	0.055966	0.192186	0.014784	0.021119
WMA_4	0.074974	0.033791	0.173178	0.040127
WMA_5	0	0	0.090813	0.097149
WMA_6	0	0	0	0

Table 5. The appraisal value.

	Appraisal value
WMA_1	0.438439
WMA_2	0.342623
WMA_3	0.440984
WMA_4	0.5
WMA_5	0.560856
WMA_6	0.5

5. Conclusions

This study proposed the machine learning and neutrosophic set model for Windows malware detection. 6 ML models are used for prediction tasks. Then the single-valued neutrosophic set is used to overcome uncertainty in the evaluation process. The EDAS method is used to select the best ML model under different matrices.

To improve detection accuracy, future studies might investigate hybrid strategies that combine many classifiers or make use of ensemble learning techniques. Furthermore, the creation of unique features catered to the traits of Windows malware may strengthen malware detection techniques even more. Our study emphasizes how crucial it is to continuously innovate malware detection techniques to protect computer systems from changing threats. We want to support further efforts to strengthen cybersecurity defenses and lessen the impact of malicious software on Windows systems by utilizing supervised machine learning techniques.

References

[1] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Novel active learning methods for enhanced PC malware detection in Windows OS," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5843–5857, 2014.

[2] L. Demetrio, S. E. Coull, B. Biggio, G. Lagorio, A. Armando, and F. Roli, "Adversarial examples: A survey and experimental evaluation of practical attacks on machine learning for Windows malware detection," *ACM Trans. Priv. Secur.*, vol. 24, no. 4, pp. 1–31, 2021.

[3] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "A systematic literature review on Windows malware detection: Techniques, research issues, and future directions," *J. Syst. Softw.*, vol. 209, p. 111921, 2024.

[4] S. Naz and D. K. Singh, "Review of machine learning methods for Windows malware detection," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, 2019, pp. 1–6.

[5] C. Ravi and R. Manoharan, "Malware detection using Windows API sequence and machine learning," *Int. J. Comput. Appl.*, vol. 43, no. 17, pp. 12–16, 2012.

[6] D. Rabadi and S. G. Teo, "Advanced Windows methods on malware detection and classification," in *Proceedings of the 36th Annual Computer Security Applications Conference*, 2020, pp. 54–68.

[7] E. Amer and I. Zelinka, "A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence," *Comput. Secur.*, vol. 92, p. 101760, 2020.

[8] X. Huang, L. Ma, W. Yang, and Y. Zhong, "A method for Windows malware detection based on deep learning," *J. Signal Process. Syst.*, vol. 93, pp. 265–273, 2021.

[9] X. Ling *et al.*, "Adversarial attacks against Windows PE malware detection: A survey of the state-of-the-art," *Comput. Secur.*, vol. 128, p. 103134, 2023.

[10] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE malware detection using ensemble learning," in *Informatics*, MDPI, 2021, p. 10.

[11] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic*, vol. 4. Prentice Hall New Jersey, 1995.

[12] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.

[13] H. Wang, F. Smarandache, Y. Zhang, and R. Sunderraman, "Single valued neutrosophic sets," *Infin. Study*, vol. 12, 2010.

[14] E. K. Zavadskas, R. Bausys, A. Kaklauskas, I. Ubarte, A. Kuzminske, and N. Gudiene, "Sustainable market valuation of buildings by the single-valued neutrosophic MAMVA method," *Appl. Soft Comput.*, vol. 57, pp. 74–87, 2017.

[15] I. Deli and Y. Şubaş, "A ranking method of single-valued neutrosophic numbers and its applications to multi-attribute decision-making problems," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 4, pp. 1309–1322, 2017.

[16] J. Ye, "Single valued neutrosophic cross-entropy for multicriteria decision-making problems," *Appl. Math. Model.*, vol. 38, no. 3, pp. 1170–1175, 2014.

Received: Nov. 5, 2024. Accepted: May 1, 2025